VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Analýza heterogenních sítí

Analysis of Heterogeneous Networks

2018 Bc. Maroš Pohančeník

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Maroš Pohančeník**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Analýza heterogenních sítí
Analysis of Heterogeneous Networks

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem práce je prostudovat oblast analýzy heterogenních informačních sítí a seznámit se se shlukovacími metodami, které se v této oblasti používají. Vybrané metody implementovat, dosažené výsledky interpretovat a vzájemně porovnat.

1. Prostudujte problematiku analýzy heterogenních sítí a shlukování v nich.
2. Vyberte vhodné datové kolekce.
3. Implemenujte vybrané algoritmy pro shlukování v heterogenních sítích.
4. Experimentujte s implementovanými metodami, interpretujte nalezené shluky.
5. Vhodně zvolte reprezentaci získaných výsledků (případně vizualizujte).

Seznam doporučené odborné literatury:

[1] Shi, C., Li, Y., Zhang, J., Sun, Y., & Philip, S. Y. A survey of heterogeneous information network analysis. IEEE Transactions on Knowledge and Data Engineering, 29(1), 17-37 (2017).
[2] Sun, Yizhou, and Jiawei Han. "Mining heterogeneous information networks: a structural analysis approach." ACM SIGKDD Explorations Newsletter 14.2, 20-28 (2013).
[3] Sun, Yizhou, et al. "Rankclus: integrating clustering with ranking for heterogeneous information network analysis." Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, ACM (2009).

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Mgr. Pavla Dráždilová, Ph.D.**

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018

doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

V Ostrave 24. apríl 2018 .....

Rád by som sa na tomto mieste poďakoval pani Mgr. Pavle Dráždilovej, Ph.D. za jej odborné rady počas vývoja a písania tejto práce. Ďalej by som sa chcel poďakovať spolužiakovi Bc. Jakubovi Piškovi za pomoc pri spracovaní databazy IMDb, mojim rodičom za podporu počas celého štúdia a v neposlednom rade mojej priateľke.

Abstrakt

Cieľom tejto diplomovej práce je zoznámiť sa s problematikou analýzy heterogénnych sietí a úlohami dolovania dat, ktoré sa v tejto oblasti výzkumu používajú. To zahŕňa: výber vhodnej dátovej sady, ktorú je možné reprezentovať heterogénnou sieťou, zabezpečenie manipulácie s týmito dátami, predstavenie možnosti výberu a filtrovania dát relevantných k analýze čo umožnuje prácu s menšou dátovou sadou, preskúmanie sémantiky meta-ciest, ktoré sa dajú z týchto dát skonštruovať. Vytvorené meta-cesty použiť k ohodnoteniu hrán v homogénnej sieti vytvorenej pomocou projekcie, alebo heterogénnej sieti s dvoma typmi objektov, reprezentovanou bipartitným grafom. Pre dané sieťové koncepty vybrať a implementovať zhlukovacie algoritmy, ktoré sú založené na rôznych princípoch a umožniť tak rôzne pohľady na tieto siete. Nakoniec použiť tieto algoritmy k analýze rôznych meta-ciest, interpretovať a vizualizovať ich výsledky.

Kľúčové slová: analýza heterogénnych sietí, dolovanie dát, zhlukovanie

Abstract

The aim of this master thesis is to introduce the heterogenous network analysis and data mining tasks, which are used in this research area. This includes: selecting suitable dataset, which could be represented as a heterogenous network, ensuring manipulation with this data, introducing possibilities of selection and filtering of a subset from the dataset, which would be relevant for the analysis purpose, and enabling by that working with smaller dataset. Exploring semantics of metapaths, which can be constructed from this data. Using the metapaths as a way of valuation of edges in homogenous network constructed via projection, or heterogenous network with two types of objects, represented by bipartite graph. Selection and implementation of clustering algorithms, which are based on different principles for these network concepts and enabling by that different views on these networks. Finally using these algorithms for analysis of different metapaths, interpreting and vizualizing their results.

Key Words: analysis of heterogeneous networks, data mining, clustering

Obsah

Zoznam použitých skratiek a symbolov

DBLP	–	Digital Bibliography & Library Project
EVD	–	Eigen Value Decomposition
HIN	–	Heterogenous Information Network
IMDb	–	Internet Movie Database
SVD	–	Singular Value Decomposition
SQL	–	Structured Query Language

Zoznam obrázkov

Zoznam tabuliek

Zoznam algoritmov

Zoznam výpisov zdrojového kódu

1 Úvod

Žijeme vo svete, v ktorom je všetko prepojené. Sociálne, prírodné a informačné systémy sú zvyčajne zložené z veľkého množstva vzájomne pôsobiacich komponentov rôznych typov. Príkladmi takých systémov sú napríklad komunikačné a výpočtové systémy, World-Wide Web, biologické siete, transportné systémy, epidemické siete, teroristické siete a mnoho ďalších. Všetky tieto systémy majú jednu spoločnú vlastnosť a tou je, že sa jedná o systémy, ktoré sa dajú reprezentovať sieťou (grafom). Jednotlivé komponenty interagujú so špecifickou množinou komponentov a tým tvoria veľkú, prepojenú a heterogénnu (viac typovú) sieť. Takéto siete alebo systémy nazývame heterogénne informačné siete. Heterogénne informačné siete sú bez pochýb všadeprítomné a tvoria dôležitú časť modernej informačnej infraštruktúry [1].

Napriek nepopierateľnej informačnej hodnote sa analýze heterogénnych informačných sietí začalo venovať až v posledných rokoch. Dovtedy sa výskum zameriaval najmä na jednotlivé komponenty týchto sietí rovnakého typu a rovnakých typov väzieb v nich. Takýmto sieťam hovoríme homogénne. Pri analýze takýchto sietí však dochádza k zanedbávaniu dôležitých informácií, ktoré heterogénna sieť ukrýva[2].

Táto diplomová práca sa zameriava na spracovanie heterogénnych dát z internetovej filmovej databázy IMDb [1], uloženie týchto dát do relačnej databázy a následnú manipuláciu s nimi. Z týchto dát je následne vytváraná sieť pomocou projekcie používateľom definovanej meta-cesty na homogénnu alebo na heterogénnu sieť s dvoma typmi objektov, ktorá je reprezentovaná bipartitným grafom. V tejto heterogénnej sieti bola vybraná a implementovaná zhluková analýza ako analytický nástroj. Kapitola 2 sa zaoberá definíciami a pojmami, ktoré sú potrebné k pochopeniu témy heterogénnych informačných sietí. Kapitola 3 popisuje implementované zhlukovacie algoritmy. V kapitole 4 je popísaná filmová databáza, ktorá je doménou heterogénnej informačnej siete tejto diplomovej práce. Kapitola 5 detailne popisuje postup, akým bolo pristupované k implementácii jednotlivých častí tejto práce. Kapitola 6 je venovaná experimentom.

V prípade bibliografickej siete je použité značenie typov objektov vychadzajúce z anglického jazyka: autor (A), článok (P), téma (T), konferencia (V). V prípade filmovej siete je taktiež použité značenie, vychádzajúce z anglického jazyka herec (A), krajina (C), režisér (D), žáner (G), jazyk (L) film (M), rok (Y). Všetky výpisy zdrojových kódov, pseudokódy a ilustrácie sú uvádzané v anglickom jazyku z dôvodu zachovania konzistencie s ostatnými publikáciami.

[1] https://www.imdb.com/

2 Heterogénne informačné siete

Mnoho aktuálneho výskumu v oblasti analýzy sietí sa venuje homogénnym sieťam, ktorých vrcholy predstavujú jeden typ objektov a hrany vzťahy medzi nimi. Avšak v reálnom svete existujú vzťahy aj medzi objektami rôznych typov a tiež vzťahy rôznych typov. Príkladom môže byť vzťah „páči sa mi“, medzi rôznymi objektami ako sú osoba a príspevok, osoba a film, prípadne vzťah „označil“, medzi osobou a fotografiou atď, v rámci sociálnych sietí. Taktiež v medicínskych dátach, kde existujú rôzne druhy objektov ako lekár, pacient, choroba, liečba a rôzne vzťahy medzi nimi.

Informačná sieť predstavuje určitú abstrakciu objektov zo skutočného sveta a interakcií medzi nimi. Osvedčilo sa, že táto abstrakcia je veľmi nápomocná nielen pri reprezentácii a ukladaní dôležitých informácii, ale taktiež predstavuje nástroj, ktorým možno získať zaujímavé informácie na základe skúmania jednotlivých vzťahov. Formálne je informačná sieť definovaná autormi článkov [3, 2] nasledovne.

Definícia 1 (Informačná sieť) *Informačná sieť je definovaná ako orientovaný graf $G = (V, E)$ s funkciou mapujúcou objekty $\varphi : V \rightarrow \mathcal{A}$ a funkciou mapujúcou hrany $\psi : E \rightarrow \mathcal{R}$. Každý objekt $v \in V$ patrí k práve jednému typu objektu z množiny typov objektov $\mathcal{A}; \varphi(v) \in \mathcal{A}$, a každá hrana $e \in E$ patrí k práve jednému typu relácii z množiny typov relácii $\mathcal{R}; \psi(e) \in \mathcal{R}$.*

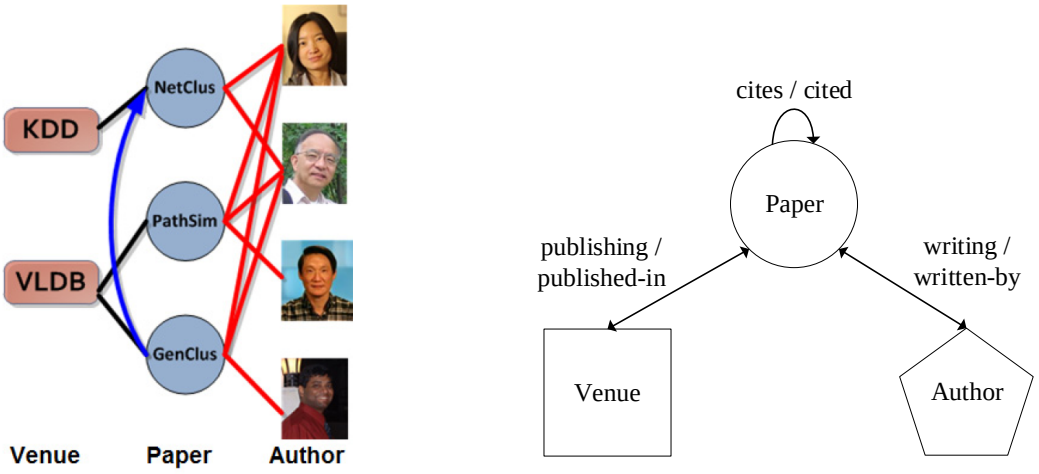
Definícia 2 (Heterogénna/homogénna sieť) *Informačná sieť je **heterogénna**, ak $|\mathcal{A}| > 1$ alebo $|\mathcal{R}| > 1$. Inak je sieť **homogénna**. Heterogénne informačné siete budeme ďalej označovať HIN.*

K tomu, aby bolo jednoduchšie porozumieť objektom a vzťahom medzi nimi v heterogénnej sieti, je potrebné poznať jej schému, prípadne informácie, ktoré sieť detailnejšie popisujú.

Definícia 3 (Schéma siete) *Schéma siete $T_G = (\mathcal{A}, \mathcal{R})$ je meta šablóna informačnej siete popísaná grafom $G = (V, E)$ s mapovacou funkciou objektov $\varphi : V \rightarrow \mathcal{A}$, a mapovacou funkciou hrán $\psi : E \rightarrow \mathcal{R}$. Graf $G(V, E)$ predstavuje orientovaný graf definovaný nad typmi objektov z $\mathcal{A}$ a hranami, ktoré predstavujú relácie z $\mathcal{R}$.*

Schéma heterogénnej informačnej siete bližšie špecifikuje isté obmedzenia na objekty a vzťahy medzi nimi. Vďaka ním sa sieť stáva viac štrukturovaná a napovedá viac o sémantike siete. Sieť, ktorá zodpovedá schéme sa nazýva inštanciou schémy. Pre hranu typu R, ktorá spája objekt typu S, s objektom typu T (značíme $S \xrightarrow{\mathrm{R}} T$), predstavuje S zdrojový typ objektu a T cieľový typ objektu, hrany typu R. Inverzná (opačná) relácia R^{-1} prirodzene predstavuje $T \xrightarrow{R^{-1}} S$. Ak R nie je symetrická tak vo všeobecnosti platí, že $R \neq R^{-1}$. Pre symetrické relácie (reprezentované neorientovaným grafom) platí, že $R = R^{-1}$.

Na rozdiel od homogénnych sietí, v heterogénnych sieťach môžu byť objekty prepojené cez rôzne cesty, ktoré majú rôzny význam. Tieto cesty sú označované ako meta-cesty.



(a) Inštancia siete DBLP

(b) Schéma siete DBLP

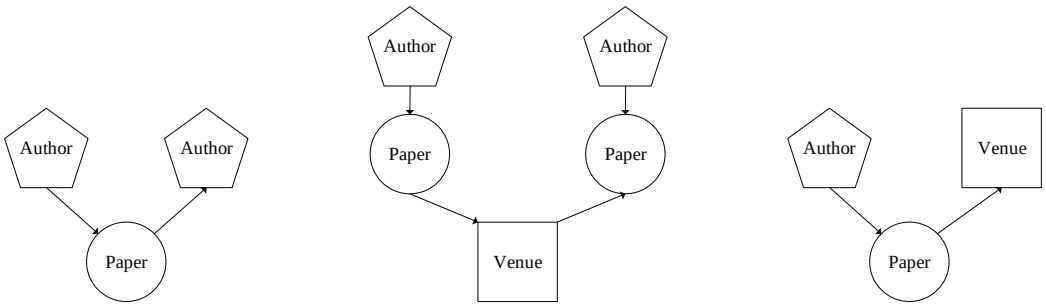
Obr. 1: Príklad heterogénnej informačnej siete nad bibliografickými dátami [2]

Definícia 4 (Meta-cesta) *Meta-cesta $\mathcal{P}$ je cesta v schéme $S = (\mathcal{A}, \mathcal{R})$ značená ako $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \ldots \xrightarrow{R_l} A_{l+1}$, čo značí zloženú reláciu $R = R_1 \circ R_2 \circ \cdots \circ R_l$ medzi objektami $A_1, A_2, \ldots, A_{l+1}$, kde $\circ$ označuje operátor skladania relácií.*

Meta-cestu je tiež možné zaznačiť pomocou typov objektov, ak medzi nimi neexistuje viacero typov relácii. Príkladom môžu byť autori, ktorí publikujú články. Túto reláciu zachytáva cesta $A \xrightarrow{písal} P \xrightarrow{bol\ napísaný} A$ skrátene možno zaznačiť ako APA. Meta-cesta $\mathcal{P}$ je symetrická ak touto cestou definovaná relácia R je symetrická. Napríklad APA (Obr. 2a) alebo $APVPA$ (Obr. 2b) atď. Meta-cesty $\mathcal{P}_1 = (A_1, A_2, \ldots, A_l)$ a $\mathcal{P}_2 = (B_1, B_2, \ldots, B_k)$ sú zreťaziteľné vtedy a len vtedy, ak A_l sa rovná B_1. Zreťazené meta-cesty značíme $\mathcal{P} = (\mathcal{P}_1, \mathcal{P}_2)$ čo znamená $(A_1 A_2 \ldots A_l B_2 \ldots B_k)$. Príkladom zreťazenia môže byť zreťazenie meta-cesty AP a PA do APA alebo APV a VPA do $APVPA$.

Tabuľka 1: Príklady meta-ciest a ich sémantického významu v bibliografickej sieti

Meta-cesta	Zápis	Sémantický význam
Autor - Článok - Autor	APA	Autori a_i a a_j sú spoluautori
Autor - Článok - Konferencia	APV	Autori a_i a a_j publikovali články na rovnakej konferencii.
Autor - Článok - Konferencia - Článok - Autor	APVPA	Autori a_i a a_j publikovali články na rovnakej konferencii.
Autor - Článok - Autor - Článok - Autor	APAPA	Autori a_i a a_j sú spoluautori, tých istých autorov.
Autor - Článok - Téma - Článok - Autor	APTPA	Autori a_i a a_j píšu o rovnakej téme.



(a) APA (b) APVPA (c) APV

Obr. 2: Ukážka meta-ciest v bibliografickej heterogénnej sieti [2]

V tabuľke 1 sú uvedené rôzne príklady meta-ciest a ich sémantických významov. Je evidentné, že každá meta-cesta skrýva iný význam. Meta-cesta APA predstavuje reláciu autorov, ktorí spolu napísali článok, zatiaľ čo meta-cesta $APVPA$ reláciu autorov, ktorí publikovali články na rovnakej konferencii. Meta-cesta môže začínať a končiť v rôznom type objektov. Napríklad cesta APV (Obr. 2c), ktorá predstavuje autorov publikujúcich články na konferenciách.

Táto sémantika meta-ciest je veľmi dôležitou vlastnosťou HIN [4]. Na základe rôznych meta-ciest majú objekty medzi sebou rôzne vzťahy, čo môže mať dopad na úlohy dolovania dát. Napríklad miera podobnosti medzi autormi na rôznych meta-cestách bude rôzna [5]. Spoluautori článkov budú mať vyššiu mieru podobnosti na ceste APA, zatiaľ čo autori, ktorí publikovali články na rovnakej konferencii budú mať vyššiu mieru podobnosti na ceste $APVPA$. Ďalším príkladom môže byť meranie dôležitosti objektov. Dôležitosť autorov v ceste APA je založená na autoroch, ktorí píšu články s mnohými ďalšími autormi, kým cesta $APVPA$ hovorí o autoroch, ktorí publikujú články na produktívnych konferenciách.

2.1 Príklady datasetov heterogénnych sietí

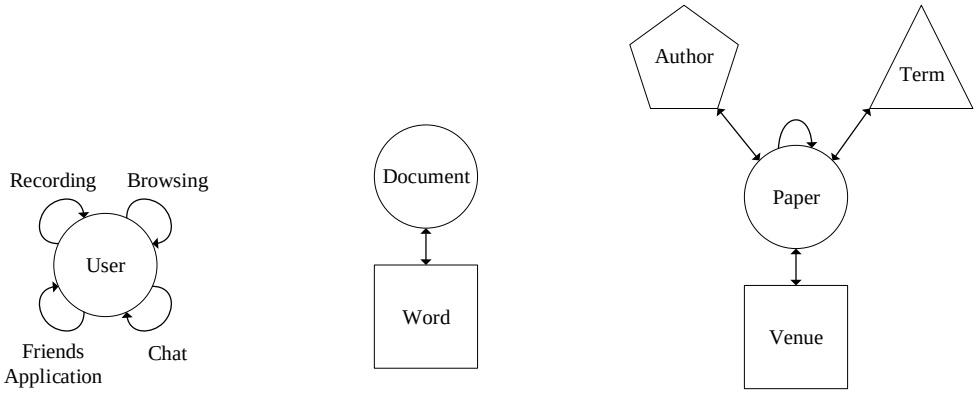
Napriek tomu, že heterogénne siete sú všadeprítomné, neexistuje mnoho štandardizovaných datasetov k štúdiu. Heterogénnu sieť však možno zostaviť z dát rôzneho zamerania ako je sociálne, vedecké, inžinierske atď. Ako príklady je možné uviesť:

Bibliografická informačná sieť. Takúto sieť môže predstavovať napríklad databáza DBLP [2]. Za objekty v tejto sieti možno považovať *článok*, *konferenciu*, *autora* a *tému*, a vzťahy *autor napísal článok*, *článok bol prezentovaný na konferencii*, *článok pojednáva o téme* a podobne.

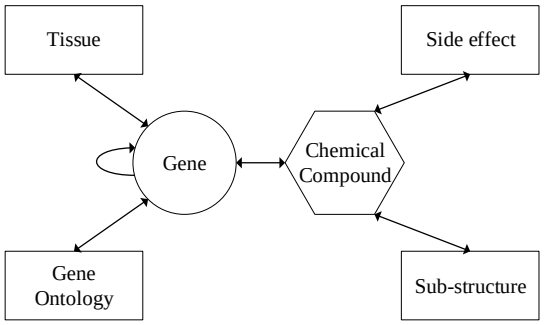
Medicínska informačná sieť. V takejto sieti je možné nájsť objekty ako *lekár*, *pacient*, *choroba*, *liečba*, a vzťahy ako pacient *navštevuje* lekára, liečba je *používaná* na chorobu atď.

[2] https://dblp.uni-trier.de/

Filmová informačná sieť. Za takúto sieť možno považovať napríklad IMDb alebo ČSFD [3]. Objekty vo filmovej sieti môžu byť *herci*, *filmy*, *režiséri* a vzťahy ako *herec hral vo filme*, *režisér režíroval film* atď.



(a) Multirelačná (b) Bipartitná (c) Hviezdicová



(d) Multihub

Obr. 3: Schémy heterogénnych informačných sietí [2]

Ďalej však treba vedieť, akou sieťou HIN vhodne reprezentovať. Tu sú uvedené niektoré príklady.

Multirelačná sieť s jedným typom objektov Tradičnou multirelačnou sieťou je sieť s jedným typom objektov a niekoľkými typmi relácii medzi objektami. Takéto siete je možné nájsť na webových stránkach ako je napríklad Facebook, Xiaonei, Twitter, Github, StackOverflow a iné. Autori [6] definovali tieto typy relácii v sieti Xiaonei (Obr. 3a).

Bipartitná sieť Bipartitná sieť je typickým príkladom HIN, používaná na analyzovanie vzťahu dvoch objektov ako napríklad *dokument-slovo* (Obr. 3b) [7, 8] alebo *používateľ-položka* [9]. Roz-

[3] https://www.csfd.cz/

šírením bipartitných sietí sú k-partitné siete, ktoré obsahujú niekoľko typov objektov a hrany existujú medzi rôznymi typmi objektov.

Sieť s hviezdicovou schémou Tento druh siete sa veľmi často vyskytuje v heterogénnych informačných sieťach. Medzi datasety, ktoré sa dajú vhodne reprezentovať hviezdicovou schémou patrí napríklad bibliografická sieť DBLP (Obr. 3c) [5], filmová sieť IMDb [10] alebo patentové dáta [11]. V prípade DBLP predstavuje ústredný prvok (hub) článok, a ostatné objekty sú jeho atribútmi. V prípade IMDb je týmto ústredným článkom film.

Sieť s viacerými rozbočovačmi Niektoré siete majú ešte komplexnejšiu štruktúru, než dokáže zachytiť sieť s hviezdicovou schémou. Tieto siete majú viac než jeden rozbočovač (hub) . Príkladom takýchto sietí sú bioinformatické dáta (Obr. 3d), kde sieť obsahuje dva rozbočovače: gén a chemickú zlúčeninu.

2.2 Príklady rôznych konceptov sietí

S nárastom analýz sociálnych sietí pribudlo aj množstvo druhov sieťových dát a s nimi aj koncepty na modelovanie týchto dát. Tu sú uvedené niektoré z nich v porovnaní s konceptom heterogénnych sietí.

Homogénna sieť Heterogénne siete obsahujú rôzne druhy vrcholov alebo hrán, zatiaľ čo homogénne siete majú len jeden typ vrcholov a hrán (Definícia 2). Homogénnu sieť možno považovať za špeciálny prípad heterogénnej siete. Heterogénnu sieť možno previesť na homogénnu sieť pomocou projekcie [12, 13] alebo ignorovaním heterogenity, čo však môže viesť k strate informácií. Algoritmy dolovania dát z homogénnych sietí ako je napríklad zhlukovanie, či ranking nie je možné priamo aplikovať na heterogénne siete.

Multirelačná sieť Multirelačné siete majú jeden typ vrcholov ale viac ako jeden typ hrán. Takže multirelačné siete sú považované za špeciálny prípad heterogénnych sietí. V článku [14] autori skúmajú možnosti predikcie hrán v takejto sieti. Autori článku [15] sa venujú detekcii komunít.

Zložená sieť V dnešnej dobe sú ľudia zapojení v niekoľkých sociálnych sieťach a vytvárajú tak zloženú sieť, kde prepojením medzi týmito sieťami sú práve ich používatelia. Interakcie používateľa v jednej sieti môžu ovplyvniť jeho správanie sa v inej sieti. Takúto dynamiku možno zachytiť pomocou modelu zloženej siete [6]. Je to taktiež špeciálny prípad heterogénnej siete. Napríklad DBLP a citačná sieť citeseerX[4].

[4] http://citeseerx.ist.psu.edu/

Komplexná sieť Komplexná sieť je sieť, ktorá má netriviálne topologické vlastnosti, ktoré sa nevyskytujú v jednoduchých typoch sietí ako sú mriežky (lattice) alebo náhodné grafy. Medzi tieto vlastnosti patrí rozloženie stupňov vrcholov s tzv. ťažkým chvostom (heavy tail), vysoký zhlukovací koeficient, štruktúry komunít a hierarchické štruktúry. Typickými komplexnými sieťami sú bezškálové siete alebo siete malého sveta. Štúdie dokazujú, že mnoho sietí z reálneho sveta sú komplexné siete ako napríklad sociálne siete, informačné siete, technologické siete, biologické siete atď [16]. Preto možno povedať, že mnoho reálnych heterogénnych sietí sú komplexné siete.

Multipartitná sieť Multipartitná sieť je heterogénna sieť, ktorá má rôzne typy objektov a hrán, avšak hrany neexistujú medzi vrcholmi rovnakého typu (Obr. 7, 9). V tejto práci je použitý prístup k selekcii dát a konštrukcii meta-ciest z článku [17]. Taktiež bola použitá nasledovná terminológia autorov tohto článku.

Definícia 5 (Doména) *Doména (domain) je množina objektov rôznych typov.*

Definícia 6 (Doménová sada) *Doménová sada (domain pool) je množina vzájomne nezávislých objektov rovnakého typu.*

Definícia 7 (Komunitná sada) *Komunitná sada (community pool) je neprázdna podmnožina doménovej sady.*

Definícia 8 (Komunita) *Komunita (community) je množina z komunitných sád, kde:*

- *Jeden typ objektu z komunitných sád je zvolený za bázovú sadu (base pool).*
- *Zvyšné typy objektov z komunitných sád obsahujú všetky objekty z doménovej sady, ktoré sú spojené s aspoň jedným objektom z bázovej sady.*

2.3 Príklady analytických metód

V porovnaní s homogénnou sieťou, heterogénna sieť je schopná zachytiť viac interakcií medzi objektami. Navyše tradičná homogénna sieť je zvyčajne tvorená z jedného zdroja dát, zatiaľ čo heterogénne siete ich môžu zlučovať niekoľko. Príkladom môžu byť služby spoločnosti Google, ako je Google search, G-mail, Google maps, Google+ atď. Všetky tieto služby, ktoré používateľ Google-u používa možno zlúčiť do heterogénnej siete a zachytiť tak všetky interakcie, ktoré používateľ vykonáva počas používania týchto služieb. Ďalším príkladom je prepojenie rôznych sociálnych sietí ako je Facebook, Instagram, LinkedIn. Každá z týchto heterogénnych sietí zachytáva určitú časť používateľových záujmov, a je možné prepojiť tieto siete vzájomne práve cez používateľa.

Autori [2] rozdelili úlohy dolovania dát v heterogénnych sieťach do nasledovných siedmich kategórií.

Meranie podobnosti Miera podobnosti predstavuje podobnosť objektov. Je základom rôznych úloh dolovania dát, napríklad pri vyhľadávaní na webe, zhlukovaní alebo v systémoch odporúčaní. Prístup založený na vlastnostiach, meria podobnosť objektov na základe ich vlastností pomocou kosínusovej podobnosti, Jaccardovho koeficientu alebo využívajúc ceny najkratšej cesty medzi uzlami.

Na rozdiel od miery podobnosti v homogénnom grafe, v heterogénnom grafe sa berie v úvahu nielen podobnosť dvoch objektov rovnakého typu, ale aj meta-cesta, ktorá tieto objekty spája. Rôzne meta-cesty majú rôzny sémantický význam a to môže viesť k rôznym výsledkom podobností.

Zhluková analýza Zhluková analýza je proces delenia objektov do zhlukov, kde zhluk predstavuje množinu objektov, ktoré sú si podobné. Typické zhlukovanie, ako napríklad k-means, je založené na vlastnostiach objektov. Mnoho štúdii bolo venovaných zhlukovej analýze v sieťových dátach. Tieto metódy však modelovali sieťové dáta ako homogénne siete. Používali rôzne miery, ako napríklad normalizovaný rez alebo modularita, na rozdelenie siete na menšie podgrafy.

V porovnaní s homogénnymi sieťami, heterogénne siete obsahujú viacero typov objektov, čo prináša nové výzvy v oblasti zhlukovania. Príkladom môže byť to, že zhluky môžu tvoriť objekty viacerých typov, ktoré spája nejaká spoločná téma. V bibliografickej sieti DBLP môže byť príkladom zhluk zameraný na databázovú oblasť, obsahujúci objekty ako sú autori, konferencie a články. Zhlukovanie v heterogénnych informačných sieťach zachováva bohatšie informácie, avšak doteraz je navrhnutých málo zhlukovacích algoritmov vhodných pre heterogénne siete. Najznámejšími sú RankClus a NetClus [18, 19].

Klasifikácia Klasifikácia predstavuje úlohu vytvárania modelu alebo klasifikátoru, na predikciu výstupnej kategoriálnej premennej tzv. triedy (class label) v analýze dát. Tradičné strojové učenie sa zameriavalo na klasifikáciu dát, ktoré spĺňali predpoklad, že sú nezávislé a rovnomerne rozdelené. Avšak v reálnych sieťových dátach hrany a vrcholy tento predpoklad nespĺňajú. Preto bolo tejto oblasti venovanej mnoho pozornosti.

Klasifikácia v heterogénnych sieťach prináša nové problémy kvôli nasledovným charakteristikám dát [20]:

- komplexná štruktúra siete - rôzne typy objektov majú rôzne rozloženie dát a rôzne typy hrán majú iný sémantický zmysel, preto nie je vhodné ich považovať za rovnocenné.
- nedostatok atribútov - nie všetky vzťahy sa dajú zachytiť pomocou atribútov. Pri transformácii informácii z hrán do atribútov často dochádza k vygenerovaniu vysoko dimenzionálneho riedkeho priestoru s tým ako narastá počet objektov.
- nedostatok výstupných premenných - klasifikátory vyžadujú dostatočný počet tréningových dát. Problémom v heterogénnych sieťach je, že nemusia byť k dispozícii všetky typy výstupnej premennej všetkých objektov na trénovanie.

Príkladom algoritmu, ktorý napriek týmto problémom dokáže prevádzať klasifikáciu v heterogénnej sieti je napríklad algoritmus GNetMine [20]. Podobne ako zhlukovanie je klasifikácia často spojená s ďalšou úlohou dolovania dát. Príkladom takejto integrácie je napríklad klasifikácia založená na rankingu RankClass [21].

Predikcia hrán Predikcia hrán predstavuje typický problém v oblasti dolovania dát založený na hranách v sieti. Ide o určovanie pravdepodobnosti, že existuje hrana medzi dvoma vrcholmi na základe pozorovania ostatných hrán a vlastností vrcholov. Je možné pozerať sa na tento problém ako na jednoduchý binárny klasifikátor, ktorý predikuje či hrana medzi dvoma objektami existuje alebo nie.

Predikcia hrán v heterogénnych sieťach prináša nové problémy. Prvým je, že predikované hrany sú rôznych typov, keďže objekty v HIN sú prepojené rôznymi typmi hrán. Druhým problémom je, že existujú závislosti medzi typmi hrán. Preto predikcia hrán v HIN musí predikovať kolektívne viacero typov hrán zohľadňujúc rôznorodé a komplexné vzťahy medzi rôznymi typmi hrán s využitím doplnkových informácií k predikcii [22].

Ranking V analýze sieti predstavuje ranking úlohu dolovania dát, ktorá vyhodnocuje dôležitosť alebo populárnosť objektu v závislosti na vybranej vyhodnocovacej funkcii. Typickými predstaviteľmi rankingových algoritmov v homogénnych sieťach je PageRank [23], ktorý vyhodnocuje dôležitosť objektov pomocou náhodnej prechádzky a HITS [24], ktorý vypočíta skóre autorít a rozbočovačov (authority and hub score).

Keďže heterogénna sieť obsahuje rôzne typy objektov, nie je možné považovať objekty za rovnocenné a miešať ich dokopy. Navyše rôzne druhy objektov a vzťahov v heterogénnej sieti majú rôzny sémantický zmysel, čo môže viesť k rôznym rankingovým výsledkom. Algoritmy, ktoré sú schopné rankingu v heterogénnych sieťach, sú napríklad Co-ranking [25], MultiRank [26], Chain of influencers [17] a iné.

Odporúčanie Systém odporúčaní navrhuje zákazníkom produkty, ktoré sú pre nich s vysokou pravdepodobnosťou vhodné. Takýto systém využíva mnoho techník ako získavanie informácii (IR), štatistiku a strojové učenie.

Komplexné informácie a bohatá sémantika heterogénnych sietí prisľubujú generovanie lepších odporúčaní. Rôzne typy objektov a vzťahov medzi nimi umožňujú efektívne spojiť všetky informácie, ktoré by mohli byť vhodne použité k odporúčaniu. Algoritmy, ktoré takéto odporúčania z heterogénnych sietí generujú, sú napríklad HeteRecom [10], ktorý využíva sémantiku meta-cesty na vyhodnotenie podobnosti medzi filmami, alebo SemRec [27], ktorý navyše využíva ohodnotenia hrán.

Fúzia informácií Informačná fúzia označuje proces zlúčenia informácií z heterogénnych zdrojov s rôznymi koncepčnými, kontextovými a typografickými reprezentáciami. Kvôli dostupnosti dát z rôznych zdrojov, sa fúzia týchto rozdistribuovaných informácií stala dôležitým problémom.

S nárastom heterogénnych sietí sa fúzia informácií rôznych HIN stala novým, no veľmi dôležitým problémom. Fúziou informácií z rôznych HIN je možné získať komplexnejšie a konzistentnejšie vedomosti o spoločných entitách, zdieľaných v rôznych sieťach.

3 Zhlukovacie algoritmy

Zhlukovanie, alebo v oblasti sociálnych sietí označované ako detekcia komunít (v prípade homogénnych „ľudských“ sietí) a v heterogénnych sieťach ako zhlukovanie, pretože dáva väčší zmysel nazývať objekty typu krajín alebo dokumentov zhlukom, je kľúčom k pochopeniu štruktúry komplexných sietí a získavaniu užitočných informácií z nich. Typicky sa delí na dve kategórie:

- detekcia neprekrývajúcich sa komunít,
- detekcia prekrývajúcich sa komunít.

Typickými predstaviteľmi algoritmov na detekciu neprekrývajúcich sa komunít, sú algoritmy založené na maximalizácii modularity (Kapitola 3.1), meraní *betweenness centrality*[5] alebo na spektre matice (Kapitola 3.3) a iné. Algoritmy, ktoré detegujú prekrývajúce sa komunity, sú založené na detekcii kľúk v grafe (Kapitola 3.2), alebo zovšeobecňujú lokálnu heuristiku metódy Louvain (NECTAR [28]) a mnoho ďalších.

Ďalší text popisuje vyššie uvedené algoritmy a ich implementáciu. Všetky tieto algoritmy hľadajú zhluky (komunity) globálne.

3.1 Louvain method

Jedná sa o metódu detekcie komunít v grafe vyvinutú na univerzite v Louvain [29]. Je založená na optimalizácii modularity grafu. Modularita Q nadobúda hodnotu z intervalu $< -1; 1 >$ a meria hustotu hrán vo vnútri komunity voči počtu hrán medzi komunitami. Skutočná optimalizácia modularity je NP-ťažký problém, preto sa využíva hltavý (greedy) prístup a jeho časová zložitosť je $O(n \log n)$. Pre vážený graf je výpočet Q definovaný:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j),$$

kde

- A_{ij} predstavuje váhu hrany medzi vrcholom i a j
- k_i predstavuje vážený stupeň vrcholu i, resp. $k_i = \Sigma_j A_{ij}$
- m je súčet váh všetkých hrán, resp. $m = \frac{1}{2}\Sigma_{ij} A_{ij}$
- c_i a c_j predstavujú komunity vrcholov i a j
- δ je Kroneckerova delta, resp. $\delta(u, v) = 1$ ak $u = v$, inak 0

[5] miera centrality grafu založená na najkratších cestách

Algoritmus 1: Louvain method

Input: Graph G

Output: List of communities

Graph $H = G$

while *Q is increasing* **do**

  Put each node i of H to its own community

  while *i has moved to other community* **do**

    `// Local modularity maximization`

    for *node i in vertices of H* **do**

      Put i to neighbor communities and calculate the increase of modularity ΔQ

      if $\Delta Q \geq 0$ **then**

        Put i to community with the highest increase

      else

        Node i stays in current community

    if ΔQ *not increasing* **then**

      break

  `// Construction of new graph`

  Initialize graph I

  Each community represents a new node of I

  Edges inside of community represents self edge (loop) with weight equal to sum of weight of edges

  Edges between the communities represents an edge between nodes of I with weight equal to sum of weight of the edges

  $H = I$

Algoritmus vo fáze lokálnej maximalizácie modularity presúva jednotlivé vrcholy do susedných komunít C a počíta nárast modularity ΔQ ako:

$$\Delta Q = \left[\frac{\sum_{in} + k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right]$$

kde

- $\sum_{in}$ - súčet váh vo vnútri C
- $\sum_{tot}$ - súčet váh hrán incidentných s vrcholmi v C
- k_i - súčet váh hrán incidentných s vrcholom i (vážený stupeň vrcholu)
- $k_{i,in}$ - je súčet váh hrán medzi vrcholom i a vrcholmi v C
- m - súčet váh všetkých hrán v grafe

Po tejto fáze algoritmus vytvorí zo vzniknutých komunít nový graf. Jednotlivé komunity predstavujú vrcholy nového grafu. Hrany vo vnútri týchto komunít sú reprezentované ako slučka

s váhou rovnou súčtu váh hrán vo vnútri komunít. Hrany medzi komunitami predstavujú hranu medzi vrcholmi nového grafu s váhou rovnou súčtu váh hrán medzi komunitami.

3.2 K-Clique percolation

Táto metóda (ďalej CPM) patrí k zhlukovým metódam na detekciu prekrývajúcich sa komunít [30]. Je založená na hľadaní všetkých maximálnych kľúk (Definícia 10).

Definícia 9 (Kľuka v grafe [31]) *Podgrafu $H \subseteq G$, ktorý je izomorfný nejakému úplnému podgrafu, hovoríme kľuka v G.*

Definícia 10 (Maximálna kľuka) *Maximálna kľuka je taká kľuka, ktorú nemožno rozšíriť pridaním ďalšieho vrcholu. Iným slovami kľuka, ktorá nie je podgrafom väčšej kľuky.*

Hľadanie maximálnych kľúk je zabezpečené pomocou algoritmu Bron-Kerbosch [32]. Jedná sa o rekurzívny algoritmus, ktorý využíva voľbu špeciálneho prvku (pivot). Algoritmus pracuje s troma množinami:

- R - množina vrcholov doteraz nájdenej kľuky,
- P - množina vrcholov, ktoré môžu patriť do kľuky,
- X - množina vrcholov, ktoré nesmú patriť do kľuky.

Na začiatku sú množiny R a X prázdne, a P obsahuje všetky vrcholy grafu. Za pivot je zvolený vrchol z množiny $P \cup X$ [33] s najväčším počtom susedov v P. Ktorákoľvek maximálna kľuka musí obsahovať pivot alebo vrcholy, ktoré s ním nie sú susediace. Z toho dôvodu stačí testovať iba tieto vrcholy ako potenciálne vrcholy, ktoré by mohli byť pridané do R. Algoritmus končí keď sú množiny P a X prázdne.

Matica C prekrývajúcich sa maximálnych kľúk predstavuje štvorcovú maticu $n \times n$, kde n je počet nájdených maximálnych kľúk. Na diagonále sa nachádza počet vrcholov v danej maximálnej kľuke a ostatných pozíciach je počet vrcholov, ktoré patria do kľuky i a zároveň aj do kľuky j.

$$C_{ij} = |cliques_i \cap cliques_j|$$

Prahovaná matica je definovaná nasledovne:

$$T_{ij} := \begin{cases} 1 & \text{ak } i = j \text{ a zároveň } C_{ij} \geq k \\ 1 & \text{ak } i \neq j \text{ a zároveň } C_{ij} \geq k \ -1 \\ 0 & \text{inak} \end{cases}$$

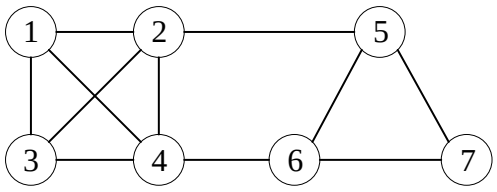
Z takto zostrojenej matice je ďalej nutné nájsť súvislé komponenty a pomocou nich následne určiť komunity v pôvodnom grafe.

Algoritmus 2: BronKerboshPivoting(R, P, X)

Input: R, P, X
Output: List of maximal cliques
if *P and X are empty* **then**
    add R to output list as a maximal clique
    return
ptemp = clone of P
pivot vertex `u` = vertex in $P \cup X$ that has the highest degree
from P remove Neighbors(`u`)
foreach *vertex v in P* **do**
    neighbors = Neighbors(v)
    rNew = $R \cup v$
    pNew = $ptemp \cap neighbors$
    xNew = $X \cap neighbors$
    BronKerboshPivoting(rNew, pNew, xNew)
    ptemp = $ptemp \cap v$
    X = $X \cup v$

Algoritmus 3: K-Clique Percolation

Input: Graph G, k
Output: List of communities
Find all maximal cliques of G
Create clique overlap matrix C
Create thresholded matrix T by thresholding C off-diagonal entries at $k-1$ and diagonal entries at k
Find connected components of T which represents communities



Obr. 4: Príklad grafu G

Podľa vyššie uvedenej definície (definícia 9) je zrejmé, že graf G na obrázku (Obr. 4) obsahuje pätnásť kľúk z toho podľa definície (definícia 10) sú štyri z nich maximálne, a to $C_1 = \{1, 2, 3, 4\}$, $C_2 = \{2, 5\}$, $C_3 = \{4, 6\}$ a $C_4 = \{5, 6, 7\}$. Matica C prekrývajúcich sa maximálnych kľúk pre

tento graf je nasledovná:

$$C = \begin{bmatrix} 4 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \\ 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 3 \end{bmatrix}$$

Prahovanie tejto matice pre parametre $k - cliques = 2, 3, 4$ dopadne nasledovne:

$$T_2 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \quad T_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Z matice T sa zistia súvislé komponenty, ktoré priamo zodpovedajú prekrývajúcim sa komunitám. Možno vidieť, že matica T_2 obsahuje iba jeden súvislý komponent, a tým pádom algoritmus vráti jednu komunitu obsahujúcu všetky vrcholy. V matici T_3 sú zrejmé dva súvislé komponenty, zodpovedajúce maximálnym kľukám C_1 a C_4, preto algoritmus vráti dve komunity obsahujúce vrcholy z C_1 a C_4. Pre maticu T_4 algoritmus vráti iba jednu komunitu obsahujúcu vrcholy z maximálnej kľuky C_1.

Táto metóda je často využívaná na detekciu prekrývajúcich sa komunít, avšak má aj isté úskalia. Pre grafy, ktoré obsahujú príliš málo kľúk alebo naopak príliš mnoho nie je vhodná. V iných prípadoch grafov zase zostane veľa vrcholov nezaradených do žiadnej komunity. Považuje sa za nepolynomiálnu metódu, avšak v realite sa ukazuje ako dostatočne efektívna.

3.3 Spektrálne zhlukovanie

Ďalšou implementovanou metódou je spektrálne zhlukovanie (Spectral clustering, ďalej SC) [34, 35]. Jedná sa o metódu, ktorá vychádza zo štandardnej lineárnej algebry a často dokáže prekonať ostatné zhlukovacie algoritmy. Algoritmus pracuje s Laplaceovou maticou a jej vlastnými číslami a vektormi. Jedna z definícii Laplacianovej matice je $L = D - A$, kde D je diagonálna matica stupňov vrcholov $D_{ii} = deg(v_i)$ a A je matica susednosti. L má na diagonále stupne jednotlivých vrcholov a mimo diagonálu sú hodnoty 0 alebo -1, ktoré reprezentujú hranu medzi vrcholmi. Príklad:

$$D - A = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 3 & -1 & -1 & 0 & -1 \\ -1 & 3 & -1 & -1 & 0 \\ -1 & -1 & 2 & 0 & 0 \\ 0 & -1 & 0 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{bmatrix}$$

V implementácii sa používa symetrická normalizovaná Laplaceova matica, ktorá je definovaná nasledovne:

$$L_{ij}^{sym} := \begin{cases} 1 & \text{ak } i = j \text{ a zároveň } deg(v_i) \neq 0, \\ -\frac{1}{\sqrt{deg(v_i)deg(v_j)}} & \text{ak } i \neq j \text{ a zároveň } v_i \text{ susedí s } v_j, \\ 0 & \text{inak.} \end{cases}$$

Normalizovaná Laplaceova matica má okrem iných, tieto vlastnosti:

- L^{sym} je symetrická,
- reálna zložka vlastných čísel je nie je negatívna,
- súčet v riadku alebo v stĺpci je 0,
- počet súvislých komponentov je rovný počtu násobností vlastného čísla $\lambda = 0$.

Z takto zostavenej matice sa vypočítajú vlastné čísla a k ním príslušné vlastné vektory. Prvých k vlastných vektorov vytvorí maticu $U^{n\times k}$, ktorá je následne normalizovaná po riadkoch. Riadky takto normalizovanej matice tvoria vstupné dáta pre algoritmus k-means, ktorý roztriedi vektory do jednotlivých zhlukov.

Algoritmus 4: Spectral clustering

Input: Graph G, k

Output: List of communities

Compute normalized Laplacian L_{sym} of G

Compute eigenvectors of L_{sym}

Take k eigenvectors and form the matrix $U^{(n\times k)}$

Form the matrix $T^{(n\times k)}$ from U by normalizing the rows to norm 1, that is set $t_{ij} = \frac{u_{ij}}{(\sum_k u_{ik}^2)^{\frac{1}{2}}}$

Cluster the rows of T by k-means

3.4 Spektrálne spoluzhlukovanie

Autorom tohto algoritmu (Spectral Co-Clustering, ďalej SCC) je Inderjit Dhillon [8], ktorý sa zaoberal zhlukovaním slov a dokumentov. Vytvoril z nich bipartitný graf a previedol tento problém na problém delenia bipartitného grafu. Na riešenie delenia bipartitného grafu použil novú metódu spektrálneho spoluzhlukovania, ktoré využíva ľavé a pravé singulárne vektory singulárneho rozkladu. Nech $A^{m\times n}$ je matica bipartitného grafu $G = (\mathcal{D}, \mathcal{W}, \mathcal{E})$, kde $\mathcal{D} = \{d_1, \ldots, d_n\}, \mathcal{W} = \{w_1, \ldots, w_m\}$ sú dve množiny vrcholov a $\mathcal{E} = \{(d_i, w_j) : d_i \in \mathcal{D}, w_j \in \mathcal{W}\}$ je množina hrán medzi dokumentami a slovami. Následne je možné zostaviť maticu susednosti $M^{(m+n)\times(m+n)}$ ako

$$M = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix},$$

ktorá je zoradená tak, že prvých m vrcholov predstavuje slová a ďalších n dokumenty. Je evidentné, že neexistujú hrany medzi vrcholmi z rovnakej partity.

Laplaceova matica $L = D - A$ bipartitného grafu je definovaná ako

$$L = \begin{bmatrix} D_1 & -A \\ -A^T & D_2 \end{bmatrix},$$

kde D_1 a D_2 predstavujú diagonálne matice stupňov vrcholov množín $\mathcal{D}$ a $\mathcal{W}$. Autor oproti spektrálnemu zhlukovaniu použil SVD rozklad normalizovanej matice $A_{norm} = D_1^{-1/2} A\, D_2^{-1/2}$. A tak namiesto počítania vlastného vektora druhého najmenšieho vlastného čísla, počíta ľavý (u) a pravý (v) singulárny vektor, ktoré odpovedajú druhému najväčšiemu singulárnemu číslu matice A_{norm}. Navyše je táto metóda omnoho efektívnejšia, keďže počíta s maticou $m \times n$ a nie $(m+n) \times (m+n)$. Na bi-modálne rozdelenie bipartitného grafu stačí vytvoriť vektor $z_2^{(m+n),1}$ a spustiť na ňom algoritmus k-means.

$$z_2 = \begin{bmatrix} D_1^{-1/2} & u_2 \\ D_2^{-1/2} & v_2 \end{bmatrix}$$

Tento algoritmus je jednoducho rozšíriteľný na k-modálne delenie grafu. Pretože tak ako druhé singulárne vektory obsahujú dostatočné informácie na bi-modálne rozdelenie grafu, tak $l = \lceil \log_2 k \rceil$ singulárnych vektorov $U_l = [u_2, u_3, \ldots, u_{l+1}]$ a $V_l = [v_2, v_3, \ldots, v_{l+1}]$ obsahujú k-modálne informácie o dátach. Preto môžme vytvoriť maticu $Z_l^{(m+n),l}$ a spustiť algoritmus k-means na nej.

$$Z_l = \begin{bmatrix} D_1^{-1/2} & U_l \\ D_2^{-1/2} & V_l \end{bmatrix}$$

Algoritmus 5: Spectral Co-Clustering

Input: Graph G, k

Output: List of communities

Compute A_{norm} of G

Compute left and right singular vector of A_{norm}

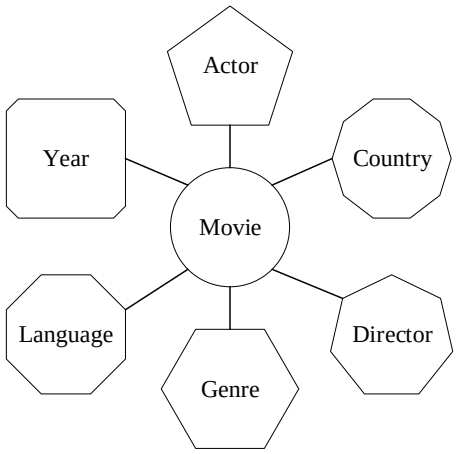
Form matrices U_l and V_l taking l vectors from computed singular vectors

Form matrix Z_l

Cluster the rows of Z_l by k-means

4 Popis IMDb databázy

Doménou tejto práce je filmová databáza IMDb. Založil ju Col Needham v roku 1990. Jedná sa o veľmi populárny medzinárodný zdroj filmového a seriálového obsahu. V roku 1998 sa stalo IMDb dcérskou spoločnosťou spoločnosti Amazon.com. K aprílu roku 2018 obsahovalo IMDb približne 4,7 miliónov titulov, 8,7 miliónov osobností[6] a viac ako 83 miliónov registrovaných používateľov. Obsah IMDb je verejne dostupný, avšak prispievanie k obsahu vyžaduje registráciu. Používatelia môžu hodnotiť film na škále od 0 do 10, toto skóre je následne prepočítané do váženého priemeru.



Obr. 5: Schéma siete IMDb (doména)

V diplomovej práci sme vytvorili databázu, ktorá sa skladá z objektov: film, herec, režisér, jazyk, rok a krajina. Z toho vyplýva aj schéma heterogénnej siete, ktorú sme použili. Všetky tieto objekty sú priamo spojené s filmom a tvoria tak hviezdicovú schému (Obr. 5).

Dáta boli získané priamo zo stránok IMDb pomocou techniky *web scraping*[7]. Boli vybrané filmy od roku 1985 po rok 2017. Ako prvé boli uložené do databázy metadáta o filmoch, ktoré obsahovali informácie či už bol film spracovaný, a prípadné chyby pri spracovaní. Následne sa prechádzali všetky zatiaľ nespracované filmy a spracovávali sa paralelne na šestnástich vláknach. Spracovanie jedného filmu predstavovalo získať povinné informácie o režisérovi, všetkých hercoch z filmu a ďalších atribútoch ako žáner, krajiny v ktorých sa film nakrúcal, dĺžka filmu, hodnotenia používateľov IMDb, rozpočet, zisk, jazyky v ktorých bol film vydaný, filmový rating MPAA[8]. Počas získavania dát boli ukladané osobné informácie o hercoch a režiséroch do spoločnej tabuľky, a v tabuľkách herec a režisér bol cudzí kľúč ako odkaz do tabuľky osobných

[6] https://www.imdb.com/pressroom/stats/
[7] web scraping - spôsob získavania dát priamo z HTML kódu
[8] MPAA - kategorizácia filmu, ktorá vyjadruje stupeň spoločenskej škodlivosti

informácií. Nakoniec sme sa rozhodli tieto dáta uložiť priamo do tabuliek hercov a režisérov. Zjednodušil sa tak spôsob filtrovania dát.

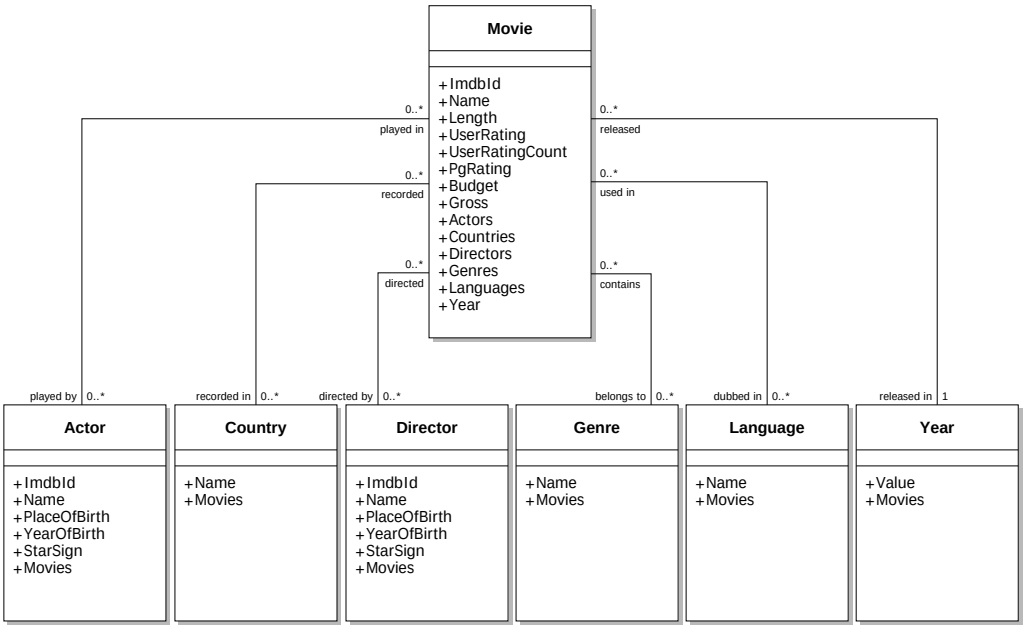
Niektoré filmy boli vypustené, pretože neobsahovali povinné atribúty, ako napríklad režiséra filmu alebo hercov a pod. Pre niektoré nepovinné atribúty, ktoré sa nepodarilo získať boli vytvorené špeciálne kategórie.

Dáta boli ukladané do relačnej databázy MySql pomocou transakcií. Z tabuľky 2 možno vidieť, že databáza obsahuje dostatočné množstvo dát na to, aby dávalo zmysel na nich aplikovať algoritmy na dolovanie dát.

Tabuľka 2: Počty typov objektov v IMDb

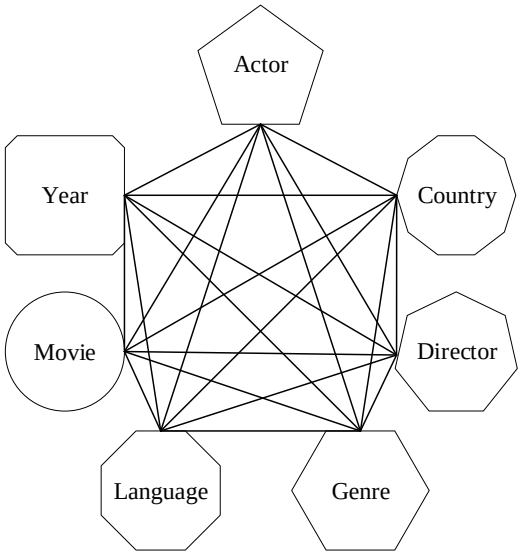
Názov tabuľky	Počet záznamov
Herci	1 485 797
Filmy	191 979
Režiséri	92 452
Krajiny	757
Jazyky	288
Roky	33
Žánre	27

Je prirodzené, že herci hrali vo viacerých filmoch a vo filmoch hralo niekoľko hercov. Taktiež je možné, že film bol režírovaný viacerými režisérmi a režiséri režírovali niekoľko filmov. To isté platí o jazyku, v ktorom bol film vydaný a naopak. Z toho vyplýva, že medzi filmom, ktorý je ústredným prvkom IMDb a ostatnými objektami budú väzobné tabuľky zachytávajúce vzťah $M : N$. Na obrázku (Obr. 6) možno vidieť pôvodný doménovy model IMDb a všetky jeho atribúty.



Obr. 6: Doménový model pôvodnej IMDb

Dodatočne boli vygenerované väzobné tabuľky medzi všetkými dvojicami objektov cez spoločný film. Pri použití týchto väzobných tabuliek dochádza k zanedbávaniu počtu filmov. Po doplnení väzobných tabuliek došlo k tomu, že schéma siete sa zmenila z hviezdicovej na úplný graf (Obr. 7). S takouto sieťou je možné ďalej pracovať ako s multipartitnou [17]. Doménový model sa tiež rozšíril, obsahuje väzby medzi všetkými dvojicami modelov. Viac sa tomuto rozšíreniu budeme venovať neskôr (Kapitola 5.1).



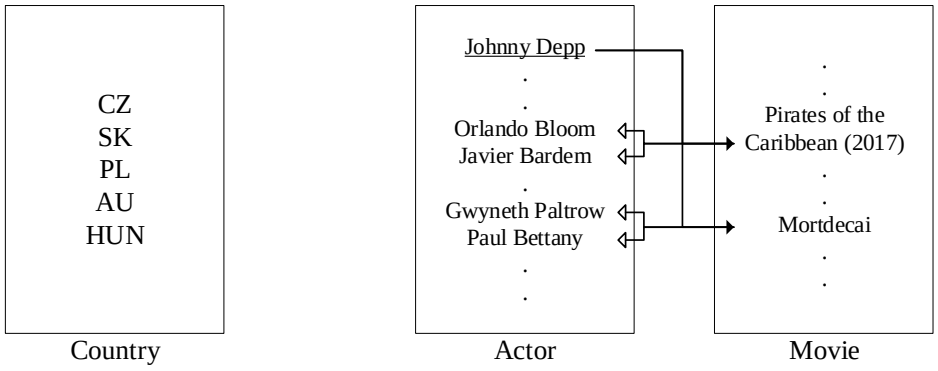
Obr. 7: Rozšírená schéma IMDb na úplný graf

4.1 Spôsoby výberu bázovej sady

Bázova sada predstavuje množinu objektov rovnakého typu, ktoré sú vybrané podľa používateľom definovaných kritérií. Umožňuje tak prácu s dátami, ktoré sú podstatne menšie a obsahujú informácie, ktoré sú relevantné voči tomu, čo chce používateľ analyzovať. Bez tohto prístupu by všetky algoritmy na analýzu siete museli bežať nad celou databázou, čo by bolo veľmi časovo náročné. Umožňujeme dva spôsoby ako špecifikovať množinu objektov v bázovej sade:

1. špecifikovať jednotlivé objekty, ktoré chceme analyzovať nad jedným typom objektov,
2. projekcia cez druhý typ objektu. Pri tejto projekcii sa využívajú vygenerované väzobné tabuľky.

Príkladom prvého spôsobu môže byť, že si vyberieme konkrétne krajiny, ktoré nás zaujímajú (Obr. 8a). Príkladom druhého spôsobu môže byť analýza režisérov, ktorí produkovali filmy v rovnakých žánroch, vtedy nám nezáleží na počte filmov. Samozrejme je možné vytvoriť takúto reláciu cez spojenie tabuliek $Director - Movie - Genre$. Keďže režisér režíroval niekoľko filmov a niektoré z nich patria do rovnakého žánru, takáto relácia hovorí koľko filmov z režisérovej tvorby patrí do dáneho žánru, alebo tiež ako veľmi bol režisér produktívny v danom žánri. Aby sme sa tejto vlastnosti zbavili, museli by sme vykonať ďalšiu databázovú operáciu `DISTINCT`. Spájanie tabuliek cez film teda reflektuje „počet filmov“. Preto môžeme použiť vopred vygenerovanú väzobnú tabuľku $Director - Genre$, ktorá túto vlastnosť (početnosť filmov) nereflektuje ale priamo hovorí, do akých žánrov patrí režisérova tvorba. Väzobná tabuľka teda predstavuje binárnu reláciu, ktorá hovorí, že režisér produkoval v danom žánri.



(a) Podmnožina jedného typu objektu

(b) Projekciou

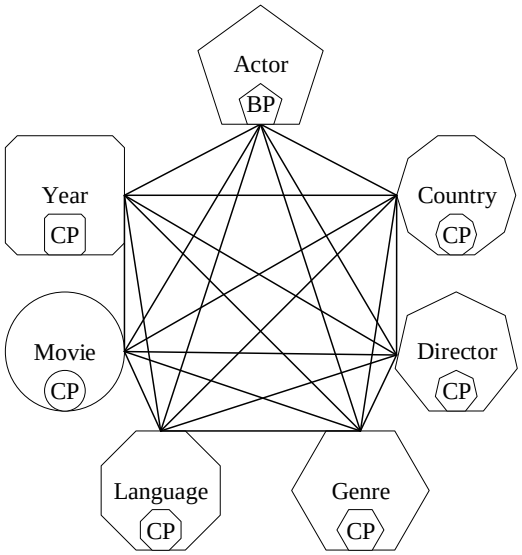
Obr. 8: Spôsoby výberu bázovej sady

Ďalším príkladom môže byť analýza všetkých hercov, ktorí hrali s nami vybraným hercom. Použijeme teda reláciu $Actor - Movie$ a vyberieme herca napríklad podľa jednoznačného identifikátora $ImbdId$. Na obrázku 8b bol zvolený herec Johnny Depp. Hrany s plnou šípkou smerujúce od tohto herca predstavujú reláciu „herec hral vo filme“. Hrany smerujúce od filmu s prázdnou šípkou zase reláciu „vo filme hrali herci“. Takouto projekciou sme získali všetkých spoluhercov Johnnyho Deppa. Pre tento konkrétny príklad bázová sada obsahuje 4468 hercov.

Po vybraní bázovej sady sa následne načítajú aj príslušné komunitné sady, kde bázovou sadou je herec Johnny Depp a jeho spoluherci, a zo zvyšných doménových sád sú vybrané príslušné komunitné sady. To zaručuje, že vždy pracujeme s dátami, ktoré je možné prepojiť pomocou akýchkoľvek meta-ciest. Tento výber bázovej sady a komunitných sád ilustruje obrázok (Obr. 9), kde zmenšené tvary predstavujú podmnožiny jednotlivých typov objektov. Skratka BP označuje bázovú sadu (Base Pool) a komunitné sady sú označené skratkou CP (Community Pool). V prípade vyššie uvedeného príkladu s Johnnym Deppom, predstavujú tieto relácie reprezentované maticami susednosti približne 2,70 GB dát pri použití hustých matíc a 6 MB pri reprezentácii dát pomocou riedkych matíc v operačnej pamäti.

4.2 Meta-cesty v IMDb

Meta-cesty sú tvorené z binárnych heterogénnych relácii takým spôsobom, že meta-cesta reflektuje počty rôznych sledov vytvorených medzi objektami prvého typu a posledného typu v meta-ceste. Ak máme $\mathcal{R}_{AM}$ binárnu heterogénnu reláciu $\mathcal{R}_{AM} \subseteq A \times M$, môžeme ju reprezentovať bipartitným grafom $G(A, M; \mathcal{R}_{AM})$. Vo všeobecnosti je tento graf neohodnotený a orientovaný (všetky hrany vedú z vrcholov v A do vrcholov v M). Matica susednosti tohto grafu je štandardne veľkosti $(|A| + |M|) \times (|A| + |M|)$, ale keďže sa jedná o bipartitný graf, tak sú jeho tri submatice nulové. Preto je možné reprezentovať tento graf jednou maticou susednosti (v litera-



Obr. 9: Výber podmnožín z doménových sád IMDb

túre označovaná ako bi-adjacency matrix), ktorá je dimenzie ($|A| \times |M|$).

$$\mathcal{R}_{AM} = \begin{bmatrix} 0 & R_{AM} \\ 0 & 0 \end{bmatrix}$$

Databázové väzobné tabuľky korešpondujú s binárnymi heterogénnymi reláciami. Pri vytváraní meta-cesty sa tieto bi-adjacency matice násobia a na konci vzniká vážená matica susednosti, ktorej hodnoty odpovedajú počtom sledov medzi prvkami v prvej a poslednej partite.

V IMDb sieti je možné pracovať s rôznymi meta-cestami a skúmať tak sieť z rôznych pohľadov. Všetky meta-cesty sú zostavované na vopred definovanej bázovej sade a príslušných komunitných sadách. Niekoľko príkladov meta-ciest je uvedených v tabuľke 3.

Tabuľka 3: Príklady meta-ciest a ich sémantického významu v IMDb sieti

Meta-cesta	Zápis	Sémantický význam
Herec - Film - Herec	AMA	Herci a_i a a_j hrali v spoločnom filme
Herec - Režisér - Herec	ADA	Herci a_i a a_j boli režírovaní rovnakým režisérom
Herec - Film - Žáner	AMG	Herec a_i hral v žánri g_j

Rozlišujeme dva druhy meta-ciest:

- symetrické - takéto meta-cesty predstavujú matice, ktoré reprezentujeme neorientovaným váženým grafom (AMA, $AMDMA$, MAM),

- nesymetrické - ďalej rozlišujeme:
  - homogénne - takéto meta-cesty predstavujú matice, ktoré by reprezentovali orientovaný vážený graf. My sme sa rozhodli zanedbať orientáciu hrán vzhľadom na implementované zhlukovacie algoritmy ($AMDA$, $ADMA$, $ADAMA$, $MDAM$),
  - heterogénne - takéto meta-cesty predstavujú matice s dimenziami $(n \times m)$, ktoré reprezentujeme ako bipartitný graf (AMD, ADM, MAD).

Uvažujme meta-cestu $AMGMA$. Táto meta-cesta je symetrická a dá sa matematicky popísať ako matica:

$$\begin{aligned} R_{AA}^{MGM} &= R_{AM} \cdot R_{MG} \cdot R_{GM} \cdot R_{MA} = \\ &= R_{AM} \cdot R_{MG} \cdot R_{MG}^T \cdot R_{AM}^T = \\ &= (R_{AM} \cdot R_{MG}) \cdot (R_{AM} \cdot R_{MG})^T = \\ &= R_{AG}^M \cdot \left(R_{AG}^M\right)^T, \ kde \\ R_{AG}^M &= R_{AM} \cdot R_{MG} \end{aligned}$$

Matica R_{AA}^{MGM} predstavuje neorientovaný, ohodnotený graf. Hodnoty matice R_{AG}^M predstavujú počet sledov z A do G, inak povedané v koľkých filmoch žánru g_j hral herec a_i. Maticu R_{AA}^{MGM} je teda možné vytvoriť pomocou troch operácii násobenia matíc $R_{AA}^{MGM} = R_{AM} \cdot R_{MG} \cdot \left(R_{AG}^M\right)^T$. Dostaneme tak graf reprezentovaný váženou maticou susednosti. Jeho vrcholy predstavujú hercov a hrany reprezentujú reláciu medzi dvomi hercami, ktorí hrali v rovnakých žánroch s váhami odpovedajúcimi počtu filmov v kombinácii s počtami žánrov, do ktorých sa tieto filmy radili.

Príkladom nesymetrickej homogénnej meta-cesty môže byť $AMDA$, ktorú reprezentujeme ako homogénnu sieť. Túto meta-cestu predstavuje matica $R_{AA}^{MD} = R_{AM} \cdot R_{MD} \cdot R_{DA}$. Zanedbanie orientácie hrán v nesymetrickej matici R je možné previesť dvoma spôsobmi:

- $R \to min(R_{ij}, R_{ji})$ - vedie k detekcii „súvislejších“ častí siete,
- $R \to max(R_{ij}, R_{ji})$ - vhodné k zhlukovaniu v neorientovanom grafe.

Príkladom nesymetrickej heterogénnej meta-cesty môže byť meta-cesta CMY, reprezentovaná maticou $R_{CY}^M = R_{CM} \cdot R_{MY}$. Táto cesta vyjadruje, ako veľmi boli krajiny produktívne v natáčaní filmov za jednotlivé roky.

5 Implementácia

Aplikácia je implementovaná v jazyku C#, používateľské rozhranie je vytvorené pomocou technológie WPF a dáta sú uložené v MySql databáze. Na prístup k dátam sa používa ORM[9], konkrétne Entity Framework Core (EF Core)[10].

5.1 Rozšírenie doménového modelu

Pôvodne hviezdicová schéma IMDb (Obr. 5) bola rozšírená na sieť, ktorej schéma tvorí úplný graf (Obr. 7). Z pohľadu implementácie to znamenalo: buď pri každom dopyte na vzťah medzi dvoma partitami, medzi ktorými neexistovala explicitná väzba, spájať tabuľky cez tabuľku filmu čo predstavuje SQL dopyt (Výpis 1), alebo rozšíriť doménový model o všetky kombinácie možných dvojíc z pôvodných siedmych typov objektov a vytvárať potom dopyty na tieto vygenerované väzobné tabuľky (Výpis 2). Pri testovaní trval SQL dopyt (Výpis 1) priemerne 2.80 sekúnd a SQL dopyt (Výpis 2) trval priemerne 1.09 sekúnd. Oba dopyty boli testované pri vyhľadávaní krajín príslušných k 4468 hercom.

Rozhodili sme sa vytvoriť väzobné tabuľky medzi každou dvojicou typov objektov a jednorazovo vygenerovať ich obsah hlavne z dôvodu, že vytváranie takýchto dopytov je jednoduchšie pomocou reflexie a vo všeobecnosti je dopyt na väzobnú tabuľku rýchlejší než dopyt, ktorý obsahuje spájanie tabuliek a následne odstránenie duplicít. Väzobné tabuľky sú naplnené dátami, ktoré vrátil dopyt (Výpis 1). Oba dopyty (Výpis 1) a (Výpis 2) vrátia rovnaký výsledok.

```
SELECT DISTINCT AM.ActorId, CM.CountryId
FROM ActorMovie AS AM JOIN CountryMovie AS CM ON AM.MovieId = CM.MovieId
WHERE AM.ActorId IN (...)
```

Výpis 1: SQL dopyt na získanie vzťahu medzi hercom a krajinami cez film

```
SELECT ActorId, CountryId
FROM ActorCountry
WHERE ActorId IN (...)
```

Výpis 2: SQL dopyt na získanie vzťahu medzi hercom a krajinami z väzobnej tabuľky

5.2 Bázová sada a prístup k databáze

Bázová sada zabezpečuje výber a filtráciu dát z databázy pre používateľa tak, aby pracoval s menším objemom dát, ktorý je možné následne spracovať do formy grafu a riešiť nad nimi úlohy dolovania dát.

[9] objektovo relačné mapovanie
[10] https://docs.microsoft.com/en-us/ef/core/

Problémom pri implementácii tohto prístupu bolo, že v čase kompilácie programu nie je známy objekt, ktorý bude predstavovať bázovú sadu. Z toho vyplýva, že taktiež nie sú známe atribúty, na základe ktorých budú dáta filtrované. Riešením tohto problému bolo použitie reflexie.

Reflexia umožňuje získať informácie o triedach a ich vlastnostiach počas behu aplikácie, a dynamicky vytvoriť ich inštancie. Vďaka tomu je možné zobraziť na používateľskom rozhraní atribúty vybranej bázovej sady a filtrovať podľa nich. Zároveň reflexia prináša možnosť výmeny domény aplikácie za inú databázu, napríklad DBLP. Pri takejto výmene by stačilo doplniť doménové modely, ktoré by modelovali túto databázu.

Metóda, ktorá využíva reflexiu k vyhľadaniu doménových modelov je nazvaná `GetModelTypes` (Výpis 3) a nachádza sa v triede `ModelReverser`. Táto metóda umožňuje v určitom mennom priestore vyhľadať všetky typy doménových modelov, ktoré implementujú rozhranie `IDbModel`. Tieto typy sa používajú na vytvorenie zoznamu doménových modelov, z ktorých si môže používateľ vyberať či už k definícii bázovej sady alebo k voľbe objektov do meta-cesty. Druhá statická metóda, ktorá sa nachádza v triede `ModelReverser` je `GetModelProperties`. Táto metóda vracia objekt, ktorý poskytuje informácie o atribútoch zvoleného doménového modelu. Na základe týchto atribútov je potom možné vytvárať reštrikcie na doménový model, ktorý predstavuje bázovú sadu.

```
public static List<Type> GetModelTypes()
{
  const string namespaceToSearch = "Imdb.Database.Model";
  return AppDomain.CurrentDomain.GetAssemblies().SelectMany(t => t.GetTypes()).
      Where(t => t.Namespace == namespaceToSearch && t.IsClass && t.
      GetInterfaces().Contains(typeof(IDbModel))).ToList();
}
```

Výpis 3: Vyhľadanie doménových modelov

Ďalším problémom bolo generovanie SQL dopytov na databázu pomocou EF Core, ktorý neumožňuje počas behu aplikácie dynamicky dosadzovať typ entity, nad ktorou bude zostavovať dopyt do databázy. Tento problém bol taktiež vyriešený reflexiou, pomocou ktorej je možné zavolať v negenerickej metóde (Výpis 4), generickú metódu (Výpis 5) a predať do nej používateľom zvolený objekt ako parameter `T`. Všetky tieto metódy sa nachádzajú v triede `QueryGenerator`.

```
public static IQueryable<IDbModel> Query(string modelName, IEnumerable<IFilter>
    filters = null)
{
  var instance = CreateInstanceOf(modelName);
  return (IQueryable<IDbModel>) typeof(QueryGenerator)
    .GetMethod("QueryGeneric")
    .MakeGenericMethod(instance.GetType())
```

```
    .Invoke(null, new object[] {filters});
}
```

Výpis 4: Volanie generickej metódy pomocou reflexie

V generickej metóde (Výpis 5) sa zostavuje lambda výraz nad generickým parametrom `T`. Lambda výraz potom možno vložiť priamo do volania metódy `Where` nad databázovým kontextom a prenechať zostavenie dopytu na databázu EF Core frameworku.

```
public static IQueryable<T> QueryGeneric<T>(IEnumerable<IFilter> filters) where
    T : class, IDbModel
{
  var parameter = Expression.Parameter(typeof(T));
  var expressionList = new List<BinaryExpression>();
  if (filters != null){ /*Creating filters and filling the expressionList*/ }
  Expression body = null;
  for (int expressionIndex = 0; expressionIndex < expressionList.Count;
      expressionIndex++){ /*Adding expressions to body*/ }
  if (body != null)
  {
    var lambda = Expression.Lambda<Func<T, bool>>(body, parameter);
    return Context.Set<T>().Where(lambda);
  }
  return Context.Set<T>();
}
```

Výpis 5: Vytváranie objektu, ktorý umožňuje prístup do databázy

Metóda (Výpis 5) obsahuje len skrátenú ukážku implementácie. Úplná implementácia metódy je v zdrojovom kóde. Na mieste prvého komentára sa nachádza cyklus, ktorý prechádza filtre nastavené používateľom a zostavujú sa výrazy, ktoré vytvárajú reštrikcie na model v generickom parametri `T`. V mieste druhého komentára sa prechádza zoznam vyššie vytvorených reštrikcií a pridáva sa do tela lambda výrazu. V prípade, že používateľ nevytvoril žiadne reštrikcie na model, metóda vráti celý set dát. Treba poznamenať, že EF Core generuje SQL dopyty až v momente, keď sa v kóde nad objektom typu `IQueryable` zavolá metóda `ToList` prípade `FirstOrDefault` a pod.

V prípade vytvárania bázovej sady pomocou projekcie cez druhý typ objektu je v triede `QueryGenerator` metóda `ExtendBasePoolViaB`, ktorá túto projekciu vykoná a následne sa rozšíri bázová sada o objekty z tejto projekcie.

5.3 Komunitné sady a prístup k databáze

Po načítaní bázovej sady následne načítame všetky relácie priamo spojené s bázovou sadou. Ak bol za bázovú sadu zvolený objekt *Actor* tak načítame relácie $Actor-Country$, $Actor-Director$, $Actor-Genre$, $Actor-Language$, $Actor-Movie$, $Actor-Year$ a následne načítame doplnkové relácie $Country-Director$, $Country-Genre$, ..., $Movie-Year$. Všetkých relácii bude na konci $\binom{7}{2} = 21$. Relácie priamo prepojené na bázovú sadu sa vytvárajú pomocou triedy `RelationshipBuilder` a jej statickej metódy `Create`. Táto metóda má tri vstupné parametre, prvé dva sú názvy partít (`partityAName`, `partityBName`) medzi ktorými má vzniknúť relácia a tretím je zoznam unikátnych databázových kľúčov partity (`partityAIds`), podľa ktorej vyhľadávame. Databázové väzobné tabuľky, ktoré tieto relácie reprezentujú majú lexikograficky usporiadané názvy aby ich bolo možné vždy jednoznačne identifikovať. Ak napríklad zostavujeme dopyt na reláciu medzi žánrom a režisérom, je táto relácia uložená v tabuľke *DirectorGenre*.

Statická metóda `Create`, v triede `RelationshipBuilder`, využíva triedu `QueryGenerator`, ktorá obsahuje metódu `ForAinAIds`. V tejto metóde sa pomocou reflexie vytvorí inštancia triedy, ktorá reprezentuje databázovú väzobnú tabuľku a pomocou reflexie sa zavolá generická metóda `ForAinAIdsGeneric`, ktorá v parametri `T` dostane inštanciu triedy väzobnej tabuľky. `ForAinAIdsGeneric` metóda je zodpovedná za vygenerovanie lambda výrazov (Výpis 6), z ktorých následne EF Core framework vygeneruje SQL dopyt na databázu (Výpis 7).

Napríklad ak bázová sada obsahuje zoznam hercov a chceme načítať všetky filmy, režisérov atď., ktorí sú priamo prepojení s nami vybranými hercami. Posielame do metódy iba zoznam databázových kľúčov z bázovej sady. Príkladom volania metódy `Create` môže byť `Create("Actor", "Movie", new[] {24, 35, 52});`, kde prvé dva parametre určujú, že má vzniknúť relácia medzi hercami a filmami a posledný parameter je zoznam databázových identifikátorov hercov z bázovej sady. Takýmto spôsobom získame z väzobnej tabuľky dvojice, ktoré určujú, v ktorom filme herec hral. K uchovaniu týchto vzťahov je implementovaná trieda `Relationship`. V nej sú uložené názvy partít, zoznamy databázových kľúčov pre obe partity, informácie o názvoch (menách) príslušných k databázovým kľúčom a matica susednosti, ktorá zachytáva vzťahy medzi partitami. Mená hercov (názvy filmov, žánrov atď.) sa z databázy získavajú ďalším dopytom na databázu podobným spôsobom podľa databázových kľúčov.

```
public static List<(int, int)> ForAinAIdsGeneric<T>(string partityA, string
    partityB, IEnumerable<int> partityAIds) where T : class, IDbManyToManyModel
{
  string selectAId = $"{partityA}Id";
  string selectBId = $"{partityB}Id";
  var parameter = Expression.Parameter(typeof(T));
  var method = partityAIds.GetType().GetMethod("Contains");
  var call = Expression.Call(Expression.Constant(partityAIds), method,
     Expression.Property(parameter, selectAId));
```

```
  var whereLambda = Expression.Lambda<Func<T, bool>>(call, parameter);
  var selectProperty = Expression.New(typeof((int, int)).GetConstructor(
    new[] {typeof(int), typeof(int)}),
    Expression.Property(parameter, selectAId),
    Expression.Property(parameter, selectBId));
  var selectLambda = Expression.Lambda<Func<T,(int, int)>>(selectProperty,
     parameter);
  return Context.Set<T>().Where(whereLambda).Select(selectLambda).ToList();
}
```

Výpis 6: Vytváranie lambda výrazov v metóde ForAinAIdsGeneric

```
SELECT ActorId, MovieId
FROM ActorMovie
WHERE ActorId IN (24, 35, 52)
```

Výpis 7: SQL dopyt vygenerovaný metódou ForAinAIdsGeneric

Keď načítavame doplnkové relácie, už poznáme oba zoznamy identifikátorov a chceme získať záznamy, ktoré zachytávajú vzťahy napríklad medzi režisérmi a krajinami, ktoré sú relevantné voči bázovej sade. Vtedy sa volá metóda `Join` z triedy `RelationshipBuilder`. V tejto metóde sa volá statická metóda `BetweenAandB` z triedy`QueryGenerator`. V tejto metóde sa opäť vytvorí pomocou reflexie inštancia triedy, ktorá reprezentuje databázovú väzobnú tabuľku a pomocou reflexie sa zavolá generická metóda `BetweenAandBGeneric`, ktorá v generickom parametri `T` dostane tento objekt. Opäť sa zostavia lambda výrazy a EF Core framework z nich vygeneruje SQL dopyt (Výpis 8). Výsledok, ktorý vráti táto metóda sa opäť uchováva ako objekty triedy `Relationship`, v tomto prípade sa už však nevykonáva ďalší dopyt do databázy k získaniu názvov krajín, žánrov atď., pretože tie už sú v programe prítomné.

Na konci tak máme v pamäti zoznam s 21 objektami typu `Relationship`, ktoré predstavujú naše komunitné sady, nad ktorými je možné budovať meta-cesty.

```
SELECT CountryId, GenreId
FROM CountryGenre
WHERE CountryId IN (2, 3, 20) AND GenreId IN (13, 15, 17, 27);
```

Výpis 8: SQL dopyt vygenerovaný metódou BetweenAandBGeneric

5.4 Používanie meta-ciest

V čase zostavovania meta-cesty je už v pamäti načítaný zoznam všetkých relácii, nad ktorými je možné meta-cestu zostaviť. Počas zostavovania sa vytvára nový zoznam relácii, ktoré zodpovedajú meta-ceste. Po definovaní meta-cesty používateľom sa overí symetrickosť meta-cesty.

V prípade symetrických meta-ciest sa prechádza polovica cesty, zvyšok cesty odpovedá transponovaným reláciám. V prípade nesymetrických meta-ciest je nutné prechádzať celú meta-cestu, navyše je dôležité rozlíšiť, či sa jedná o analýzu homogénnej alebo heterogénnej siete.

Relácie, ktoré sú načítané po fáze vytvárania komunitných sád, sú pomenované v lexikografickom poradí, avšak relácie v meta-ceste môžu byť v rôznych poradiach. Je preto treba dohľadať v zozname relácii reláciu, ktorá odpovedá relácii v meta-ceste a v prípade, že v meta-ceste je relácia opačná je nutné celú reláciu transponovať. Príkladom je meta-cesta CGD, ktorá obsahuje relácie $Country-Genre$ a $Genre-Director$. Práve druhá relácia nie je v lexikografickom poradí a v zozname relácii je už načítaná ako $Director - Genre$, takže túto reláciu transponujeme a pridáme do relácii patriacich k definovanej meta-ceste.

Po tom, čo sa zostaví zoznam relácii zodpovedajúci meta-ceste, sa následne vytvára vážená matica susednosti. Tá je v prípade symetrickej meta-cesty vytvorená prenásobením matíc tak, ako bolo popísané v kapitole 4.2. V prípade nesymetrických ciest rozlišujeme či meta-cesta začína a končí v rovnakom type objektu (homogénna sieť) alebo nie (heterogénna sieť). Ak sa jedná o homogénnu sieť je potrebné symetrizovať výslednú váženú maticu susednosti ako bolo popísané v kapitole 4.2.

5.5 Vytváranie siete z meta-cesty

V aplikácii je použitá knižnica `QuickGraph`[11], ktorá zabezpečuje objektovú reprezentáciu grafu. K vytvoreniu homogénneho grafu sa používa horná trojuholníková časť váženej matice susednosti. K vytvoreniu heterogénnej siete reprezentovanej bipartitným grafom sa prechádza celá vážená matica susednosti.

5.6 Zhlukovanie

Po zostavení grafu je možné začať s analýzou siete. Aplikovanie algoritmov je rozdelené na tri fázy, inicializáciu algoritmu, samotné zhlukovanie a vyhodnotenie kvality nájdeného zhlukovania.

Inicializácia zabezpečuje predspracovanie dát, ktoré algoritmy potrebujú k samotnému zhlukovaniu. V prípade spektrálneho zhlukovania sa v tejto fáze prevádza spektrálny rozklad matice. Spektrálne spoluzhlukovanie vykonáva singulárny rozklad matice a pri K-Clique Percolation dochádza k hľadaniu maximálnych kľúk a zostavovaniu matice prekrývajúcich sa kľúk. Tieto dáta nie sú priamo ovplyvňované vstupnými parametrami algoritmu a nemenia sa až kým nedôjde k zmene analyzovanej siete.

Druhá fáza pracuje so vstupnými parametrami algoritmu a dochádza k samotnému zhlukovaniu. V prípade spektrálneho zhlukovania aj spektrálneho spoluzhlukovania je v experimentoch použitá usporiadaná trojica argumentov ($k - means$, $k - iterations$, $k - vectors$). Parameter $k-means$ určuje koľko zhlukov má algoritmus k-means v dátach hľadať. Parameter $k-iterations$

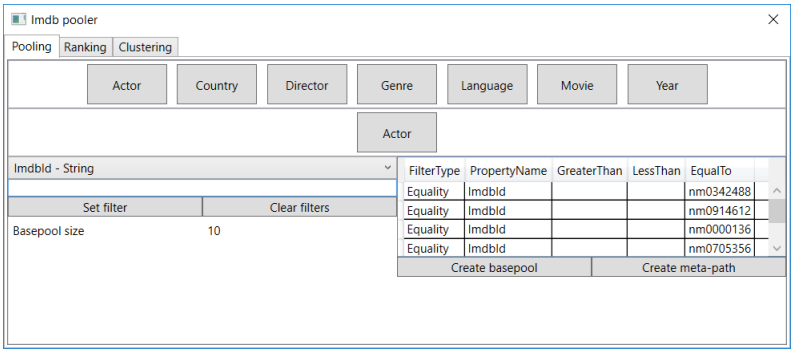
[11] https://www.nuget.org/packages/QuickGraph/

určuje koľkokrát sa algoritmus k-means zopakuje, a z týchto iterácii sa vyberie zhlukovanie s najmenšou chybou. Parameter $k - vectors$ predstavuje počet pravých a ľavých vlastných vektorov, ktoré sa použijú na zostavenie matice $U^{n \times k}$, ktorá je vstupom pre algoritmus k-means. Pri spektrálnom spoluzhlukovaní určuje parameter $k - vectors$ počet pravých a ľavých singulárnych vektorov. Tento parameter sa prepočítava na parameter l ako bolo uvedené v kapitole 3.3. Z týchto vektorov sa následne zostavuje matica $Z_l^{(m+n),l}$, na ktorej sa spúšťa algoritmus k-means. Pre algoritmus K-Clique Percolation predstavuje parameter $k - cliques$ maximálny počet vrcholov v kľuke.

Fázu zhlukovania je možné opakovať niekoľkokrát s rôznymi nastaveniami parametrov. Po skončení tejto fázy sa vyhodnocuje kvalita nájdeného zhlukovania pomocou modularity (Q). Následne je možné výsledky zhlukovania exportovať vo formáte `GEXF`[12] a zobraziť v programe Gephi[13]. Napriek tomu, že formát `GEXF` umožňuje nastaviť rôzne tvary vrcholov, Gephi nevie zobrazovať iný tvar ako kruh. Pri bipartitných grafoch rozlišujeme typy objektov pomocou záporného databázového identifikátora. Navyše až na bipartitný graf, ktorý obsahuje vrcholy hercov a režisérov, je možné rozlíšiť typy objektov podľa názvov. V prípade prekrývajúcich sa zhlukov je vrchol, patriaci do tohto prekryvu, označený farbou vypočítanou ako priemer farieb prekrývajúcich sa zhlukov (Obr. 13, 22, 28).

5.7 Používateľské rozhranie

Hlavná obrazovka je zameraná na definovanie a vytvorenie bázovej sady a komunitných sád. Ako môžeme vidieť na obrázku 10, v najvyššej časti sa nachádzajú všetky typy objektov v IMDb. Jednoduchým kliknutím si daný objekt zvolíme (v prípade projekcie volíme dva typy objektov) a nastavíme nami požadované filtre. Na obrázku 10 je špecifikovaných desať filtrov, ktoré určujú hercov priamo podľa atribútu `ImdbId`. Po definovaní filtrov, stačí kliknúť na tlačidlo `Create basepool` a počkať, kým aplikácia vyberie z databázy všetky relevantné dáta vrátane komunitných sád.

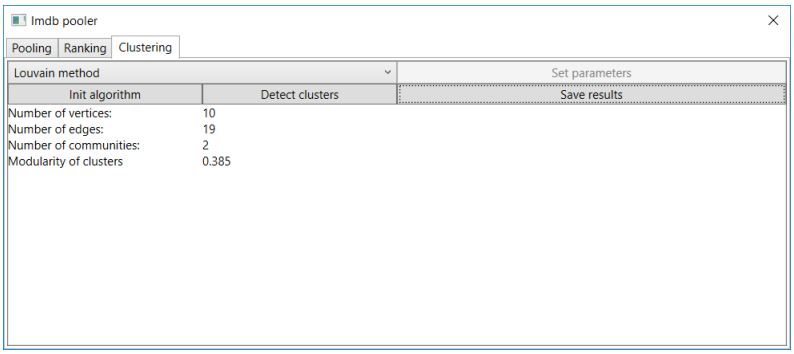


Obr. 10: Úvodná obrazovka aplikácie

[12] https://gephi.org/gexf/format/
[13] https://gephi.org/

Po tom, čo sú načítané všetky dáta z databázy, môžeme začať zostavovať meta-cesty rovnakým spôsobom akým sme definovali aj bázovú sadu. Objekt z meta-cesty odstránime kliknutím naň. Po zostavení nami požadovanej meta-cesty, klikneme na tlačidlo `Create meta-path`. Po vykonaní všetkých operácii a zostavení grafu je možné prejsť na obrazovku zhlukovania.

Na tejto obrazovke si vyberieme algoritmus, ktorým chceme sieť zhlukovať. Po vybraní algoritmu klikneme na tlačidlo `Init algorithm`, ktoré inicializuje algoritmus. Následne nastavíme vstupné parametre algoritmov v dialógovom okne, ktoré sa otvorí po kliknutí na tlačidlo `Set parameters`. V tejto fáze môžeme kliknút na tlačidlo `Detect clusters`. Po skončení zhlukovania sa zobrazí počet nájdených zhlukov. Následne môžme kliknúť na tlačidlo `Save results`, ktoré otvorí dialógové okno a umožní nám uložiť súbor vo formáte `GEXF`. Tento súbor je potom možné otvoriť pomocou programu Gephi.



Obr. 11: Obrazovka zhlukovania

6 Experimenty

V tejto časti sú uvedené experimenty nad projekciami heterogénnych sieti na homogénne a na heterogénnych sieťach reprezentovaných bipartitnými grafmi. Bázové sady boli volené zámerne menšie, aby bolo možné prezentovať výsledky zhlukovania graficky.

6.1 Analýza bázovej sady hercov

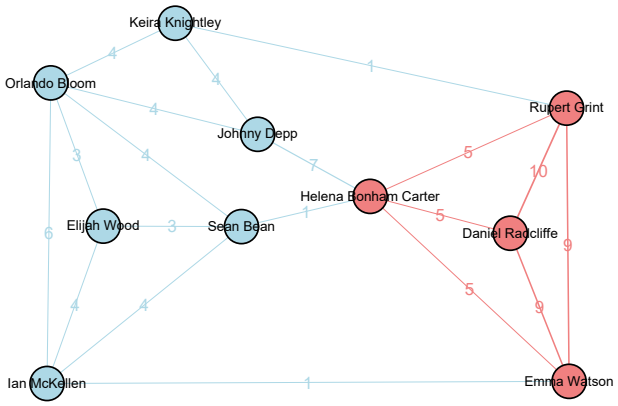
Bázová sada obsahuje desať nami vybraných hercov: Ian McKellen, Johnny Depp, Sean Bean, Helena Bonham Carter, Elijah Wood, Keira Knightley, Orlando Bloom, Daniel Radcliffe, Rupert Grint a Emma Watson.

Projekcie na homogénne siete

V tejto časti sú experimenty nad meta-cestami, ktoré začínajú a končia v type objektu herec.

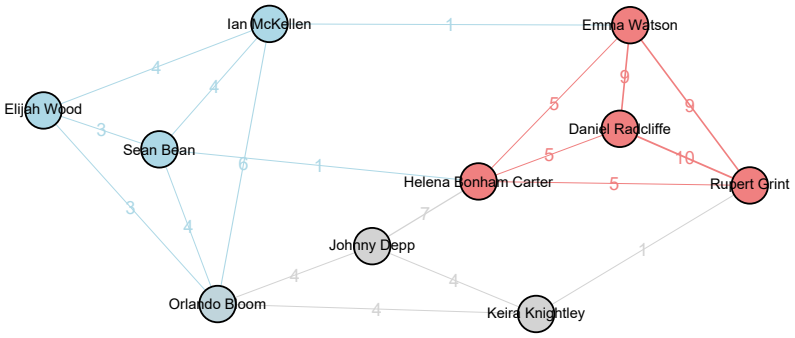
Meta-cesta AMA Táto meta-cesta predstavuje váženú maticu R_{AA}^{M}, ktorá zohľadňuje vzťahy medzi hercami, ktorí spolu hrali vo filmoch.

Algoritmus Louvain v takejto sieti detegoval dve komunity (Obr. 12). Komunitu vyznačenú červenou farbou dobre reprezentuje film Harry Potter. Druhú komunitu zase filmy Pán Prsteňov a Piráti z Karibiku.



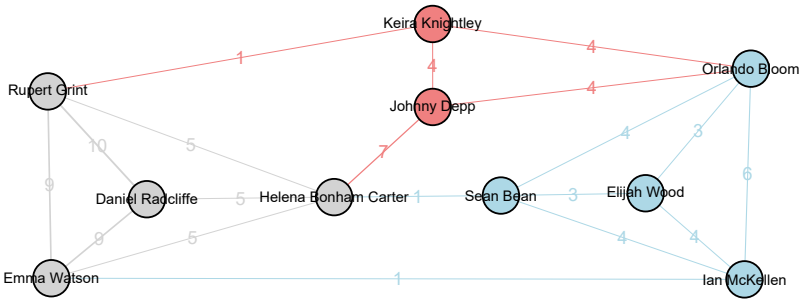
Obr. 12: Meta-cesta AMA, algoritmus Louvain, Q = 0,385

Algoritmus CPM s parametrom $k - cliques = 3$ detegoval tri komunity oproti metóde Louvain, z toho dve komunity (modrá a šedá) sa prekrývajú cez herca Orlanda Blooma (Obr. 13). Po zvýšení parametru $k - cliques = 4$ dopadlo zhlukovanie rovnako ako spektrálne zhlukovanie (Obr. 14). Herci Johnny Depp a Keira Knightley boli zaradení do vlastnej komunity.



Obr. 13: Meta-cesta AMA, algoritmus CPM (3), Q = 0,344

Spektrálne zhlukovanie s parametrami (2, 5, 3) dopadlo rovnako ako metóda Louvain (Obr. 12). Pri parametroch (3, 10, 3) zhlukovanie dopadlo obdobne ako na obrázku 14, iba herec Orlando Bloom bol zaradený do červeného zhluku. Po pridaní ďalšieho vlastného vektoru (3, 10, 4) rozdelilo zhlukovanie hercov do troch komunít (Obr. 14).



Obr. 14: Meta-cesta AMA, algoritmus SC (3, 10, 4), Q = 0,383

V zhlukovaní pri meta-ceste AMA dosiahli najlepšej modularity algoritmus Louvain, ktorý

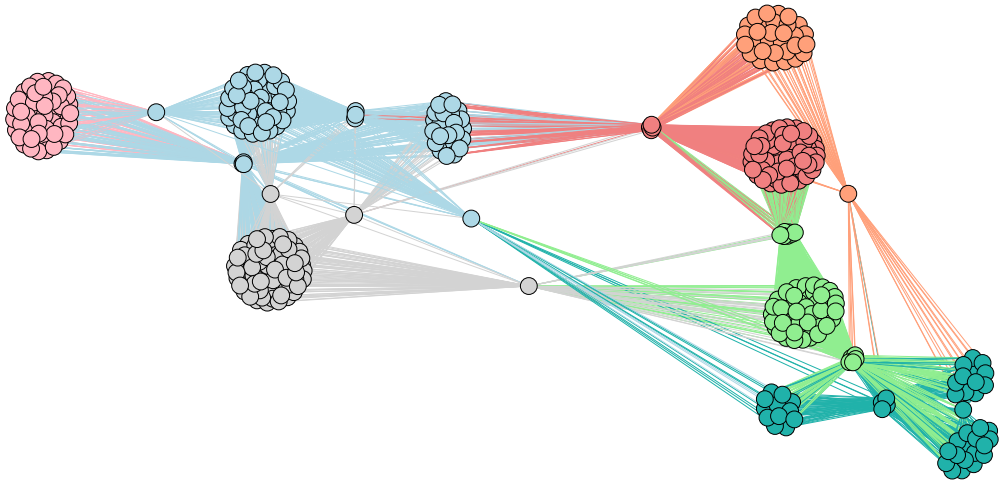
túto hodnotu maximalizuje a taktiež algoritmus spektrálne zhlukovanie. Spektrálne zhlukovanie s niektorými parametrami (Tabuľka 4) a CPM dosiahlo pre iné rozdelenie vrcholov podobne vysokej modularity. Preto sme ako ukážku vybrali (Obr. 14).

Tabuľka 4: Kvalita zhlukovania v AMA pre rôzne parametre SC, k-iterations = 100

k-means / k-vectors	2	3	4	5	6	7	8	9	10
2	0,346	0,385	0,385	0,385	0,385	0,293	0,211	0,117	0,017
3	0,383	0,344	0,383	0,231	0,299	0,130	0,094	0,156	-0,111
4	0,346	0,286	0,284	0,216	0,273	0,276	0,273	0,030	0,037
5	0,249	0,223	0,226	0,220	0,172	0,220	0,134	-0,041	0,04
6	0,187	0,187	0,163	0,163	0,196	0,163	0,052	-0,067	-0.002
7	0,125	0,117	0,128	0,139	0,139	0,139	0,083	-0,108	0,013

Meta-cesta MAM Táto meta-cesta predstavuje váženú maticu R^A_{MM} filmov, ktorej hodnoty značia počet spoločných hercov medzi filmami. Algoritmus Louvain detegoval sedem zhlukov. Na obrázku 15 je vyznačený modrou farbou zhluk, ktorý obsahuje filmy ako trilógia Pán Prsteňov, tetralógia Hobit ale aj Láska a iné pohromy, Zulu, atď. Tento zhluk predstavuje filmy Iana McKellena a Orlanda Blooma. Červený zhluk obsahuje filmy ako Piráti z Karibiku, Nožnicovoruký Edward alebo Mortdecai. Všetko sú to filmy, v ktorých hral herec Johnny Depp. Ružový zhluk obsahuje filmy Elijaha Wooda, šedý zhluk zase filmy Seana Beana. Svetlo zelený zhluk obsahuje filmy ako Alica v krajine zázrakov, Charlie a továreň na čokoládu či Mŕtva nevesta Tima Burtona. Tieto filmy sa vzťahujú k herečke Helena Bonham Carter. Tmavo zelený zhluk obsahuje filmy ako Harry Potter, Kríž Cti, Impérium, Noe, atď. Sú to filmy hercov: Daniel Radcliffe, Rupert Grint a Emma Watson. Oranžový zhluk predstavuje filmy herečky Keiry Knightley.

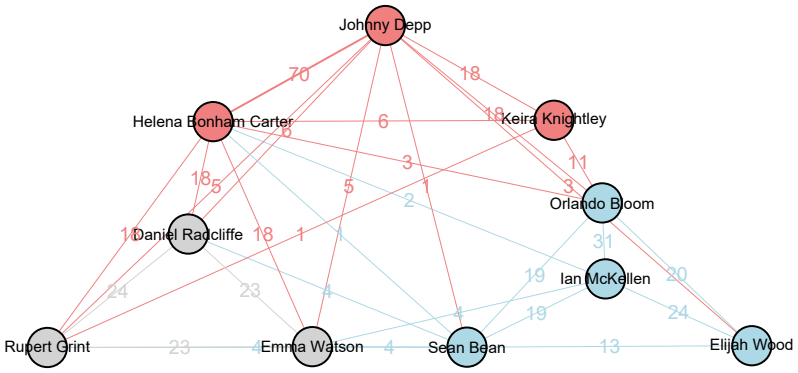
Spektrálne zhlukovanie s parametrami (7, 5, 7) nad touto sieťou dopadlo obdobne ako algoritmus Louvain s výslednou modularitou Q = 0,695.



Obr. 15: Meta-cesta MAM, algoritmus Louvain, Q = 0,696

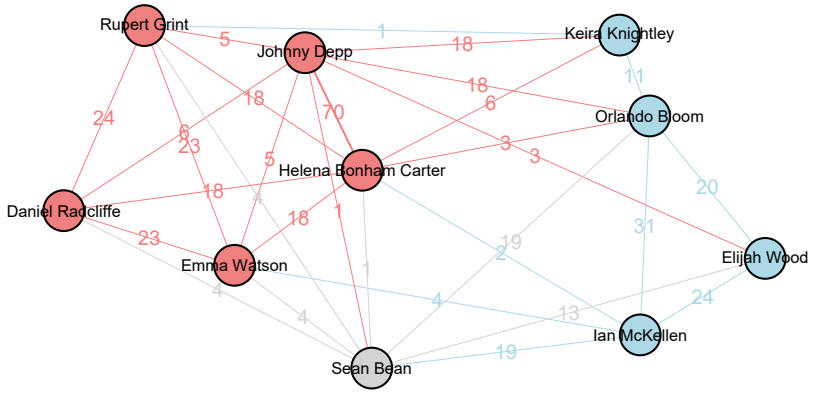
Meta-cesta AMDMA Táto meta-cesta predstavuje váženú maticu R_{AA}^{MDM} hercov, ktorí boli režírovaní spoločným režisérom zohľadňujúc počty filmov. Johnny Depp a H. B. Carter majú medzi sebou veľmi silnú väzbu. Je to spôsobené tým, H. B. Carter a Johnny Depp hrali spolu vo väčšine filmov, ktoré režíroval Tim Burton.

Na obrázku (Obr. 16) môžeme vidieť, ako dopadlo zhlukovanie nad bázovou sadou pomocou algoritmu Louvain. Algoritmus K-Clique Percolation nie je vhodný na analýzu tejto siete. Jeho výsledkom je vždy jedna komunita všetkých hercov alebo žiadna. Je to z toho dôvodu, že táto sieť je príliš prepojená.

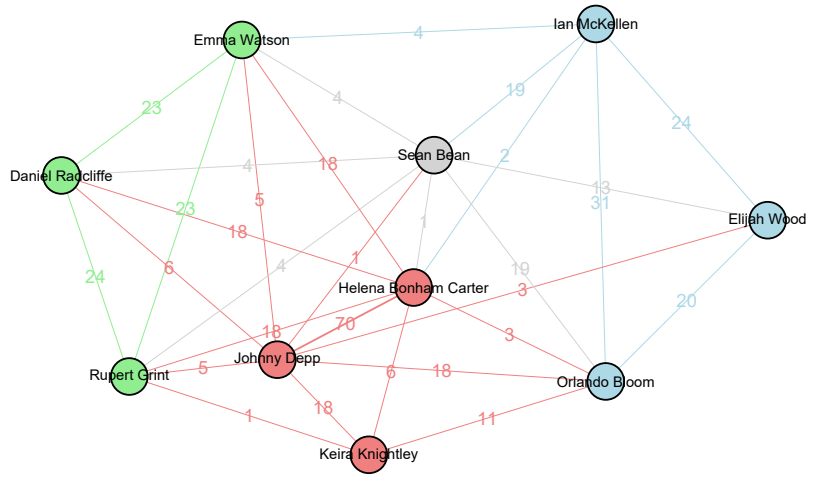


Obr. 16: Meta-cesta AMDMA, algoritmus Louvain, Q = 0,358

Spektrálne zhlukovanie s parametrami (2, 10, 3) v tejto sieti rozdelilo hercov rovnako ako algoritmus Louvain v prípade meta-cesty AMA (Obr. 12). Pri parametroch (3, 10, 3) boli herci rozdelení do troch komunít ako možno vidieť na obrázku 17. Po zvýšení parametru $k-vectors = 4$ sa kvalita zhlukovania ešte zvýšila a výsledky dopadli rovnako ako pri algoritme Louvain (Obr. 16). Po zvýšení počtu zhlukov, do ktorých mali byť herci zaradení dopadlo zhlukovanie tak, ako možno vidieť na obrázku 18.



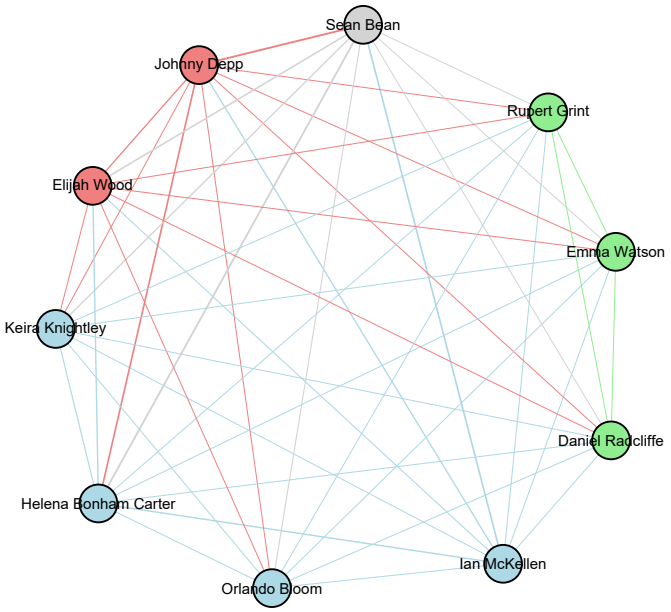
Obr. 17: Meta-cesta AMDMA, algoritmus SC (3, 10, 3), Q = 0,248



Obr. 18: Meta-cesta AMDMA, algoritmus SC (4, 10, 4), Q = 0,281

Meta-cesta AMGMA Táto meta-cesta predstavuje váženú maticu R_{AA}^{MGM} hercov, ktorí hrali vo filmoch s rovnakým žánrom zohľadňujúc aj počty týchto filmov. Z tejto relácie bola vytvorená homogénna sieť, ktorá bola následne analyzovaná. Algoritmus spektrálne zhlukovanie bol najvhodnejší, ostatné algoritmy detegovali len jednu komunitu, čo vychádza z ich podstaty. Na obrázku 19 môžme vidieť výsledok algoritmu spektrálne zhlukovanie. Algoritmus rozdelil hercov

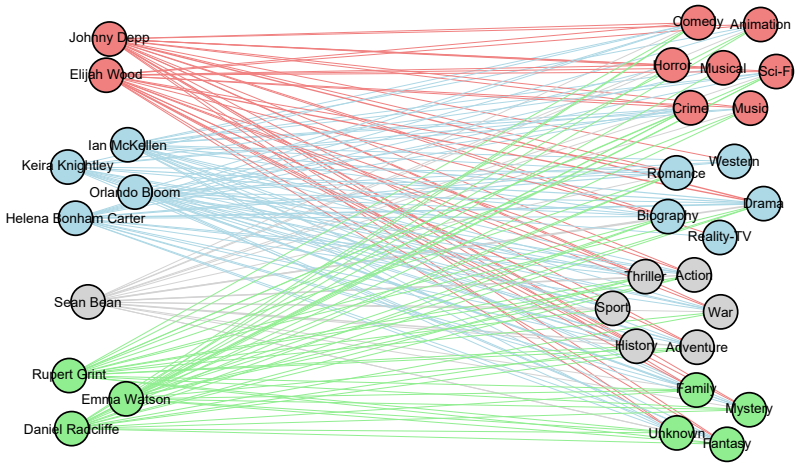
do štyroch komunít, následne môžeme porovnať výsledok zhlukovania s výsledkom zhlukovania nad meta-cestou AMG (Obr. 20). Váhy hrán nezobrazujeme kvôli prehľadnosti.



Obr. 19: Meta-cesta AMGMA, algoritmus SC (4, 10, 4), Q = -0,087

Heterogénne siete reprezentované bipartitným grafom

Meta-cesta AMG Meta-cesta AMG predstavuje váženú maticu R_{AG}^{M}, ktorú možno reprezentovať bipartitným grafom, kde jeden typ vrcholov sú herci a druhým sú žánre. Hrany v grafe sú ohodnotené, avšak kvôli prehľadnosti ich váhy nezobrazujeme. Algoritmus spektrálne spoluzhlukovania s parametrami (4, 10, 5) rozdelil vrcholy v bipartitnom grafe do štyroch zhlukov (Obr. 20).



Obr. 20: Meta-cesta AMG, algoritmus SCC (4, 10, 5), Q = 0,129

6.2 Analýza bázovej sady filmov

Za bázovú sadu sme zvolili šestnásť filmov: trilógiu Pán Prsteňov, pentalógiu Piráti z Karibiku a oktalógiu Harry Potter.

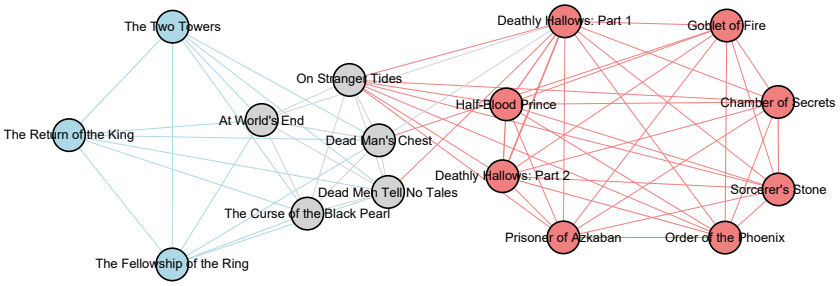
Projekcie na homogénne siete

Meta-cesta MAM Táto meta-cesta predstavuje váženú maticu R^A_{MM}, ktorú reprezentujeme homogénnou sieťou filmov. Väzby medzi filmami sú ohodnotené počtom spoločných hercov, kvôli prehľadnosti váhy nie sú na obrázkoch zobrazené.

Algoritmus Louvain detegoval v sieti tri zhluky, ktoré presne zodpovedajú bázovej sade (Obr. 21). Modrý zhluk zodpovedá trilógii Pán Prsteňov, šedý zhluk filmom Piráti z Karibiku a červený zhluk filmom Harry Potter. Rovnaké zhluky detegoval aj algoritmus SC (3, 10, 3). Ostatné výsledky s inými parametrami spektrálneho zhlukovania možno vidieť v tabuľke 5.

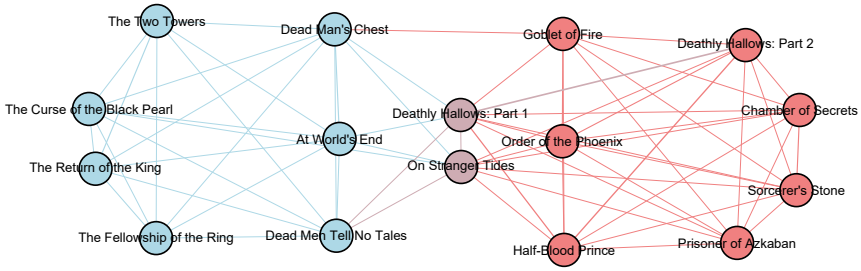
Tabuľka 5: Kvalita zhlukovania v MAM pre rôzne parametre SC, k-iterations = 100

k-means / k-vectors	2	3	4	5	6	7	8	9	10
2	0,341	0,341	0,341	0,341	0,341	0,341	0,121	0,341	0,206
3	0,327	0,353	0,353	0,353	0,353	0,353	0,198	0,243	0,135
4	0,335	0,335	0,323	0,323	0,287	0,335	0,192	0,205	0,110
5	0,318	0,318	0,318	0,318	0,278	0,276	0,198	0,148	0,140



Obr. 21: Meta-cesta MAM, algoritmus Louvain, Q = 0,353

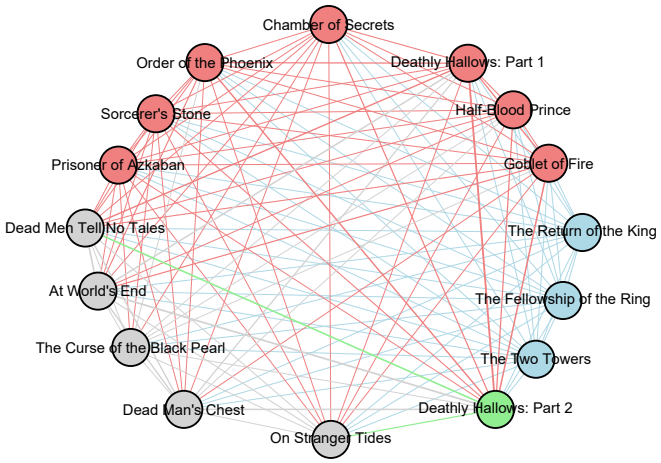
Algoritmus CPM rozdelil sieť na dva zhluky, ktoré sa prekrývajú filmami Harry Potter a Dary smrti I a Piráti Karibiku 4: V neznámych vodách (Obr. 22).



Obr. 22: Meta-cesta MAM, algoritmus CPM (4), Q = 0,313

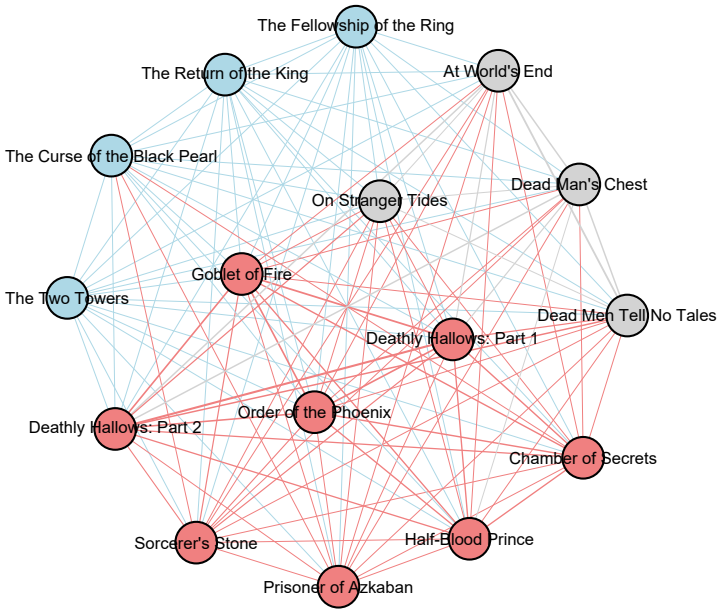
Meta-cesta MAGAM Táto meta-cesta spája filmy cez žánre, v ktorých herci pôsobia. Predstavuje váženú maticu R_{MM}^{AGA}. Metódy Louvain a CPM nie sú vhodné pre túto sieť, je príliš hustá

(z tohto dôvodu nezobrazujeme váhy hrán). Avšak algoritmom spektrálne zhlukovanie je možné zhlukovať aj v takejto sieti. Ten pri parametroch (3, 10, 3) rozdelil filmy do troch zhlukov, ktoré zodpovedali trom filmom z bázovej sady (resp. všetkým ich častiam). Po zvýšení parametru $k - vectors = 4$ priradila poslednú časť oktalógie Harry Potter k filmom trilógie Pán Prsteňov. Po zvýšení parametru $k - means = 4$ film Dary smrti II vytvoril samostatný zhluk (Obr. 23).

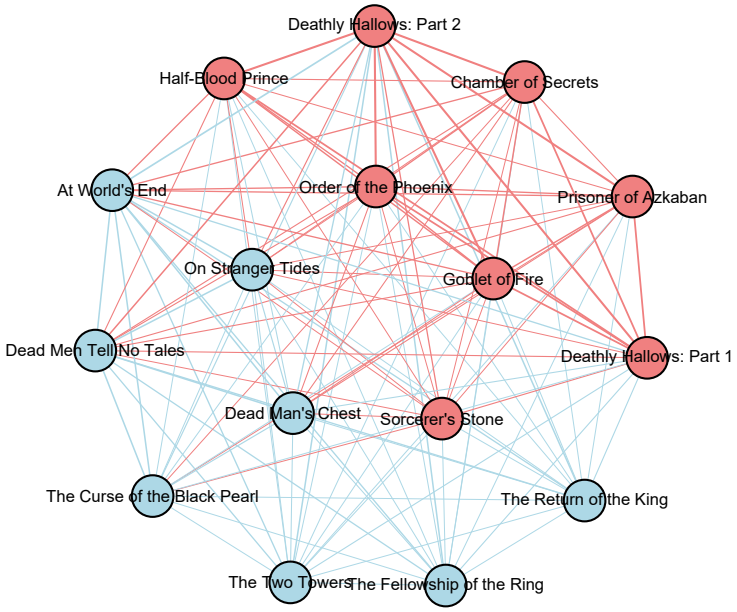


Obr. 23: Meta-cesta MAGAM, algoritmus SC (4, 10, 4), Q = -0,039

Meta-cesta MAGM Táto nesymetrická meta-cesta bola prevedená na homogénnu sieť symetrizáciou (Kapitola 4.2). Po symetrizácii pomocou minima detegoval algoritmus Louvain tri zhluky (Obr. 24a). Zaujímavé je, že zaradil jednu časť z pentalógie Piráti z Karibiku, medzi filmy Pán Prsteňov. Po symetrizácii pomocou maxima detegoval algoritmus dva zhluky (Obr. 24b). Jeden obsahuje všetky časti oktalógie Harry Potter a druhý zhluk zvyšné filmy.



(a) Symetrizácia pomocou minima, Q = 0,010



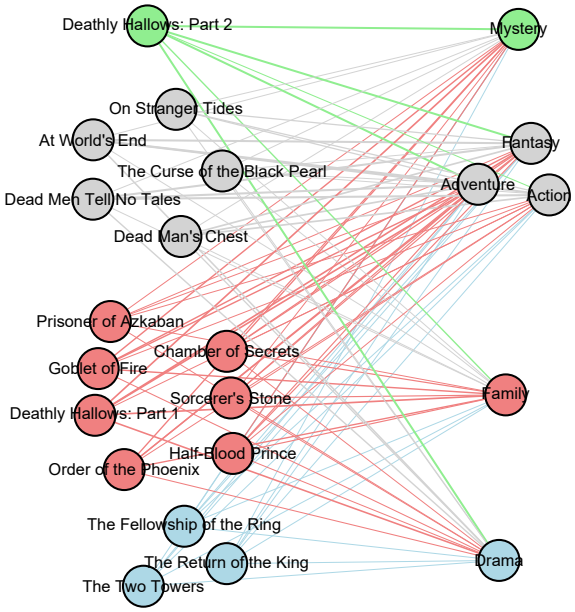
(b) Symetrizácia pomocou maxima, Q = 0,015

Obr. 24: Meta-cesta MAGM, algoritmus Louvain

Heterogénne siete reprezentované bipartitným grafom

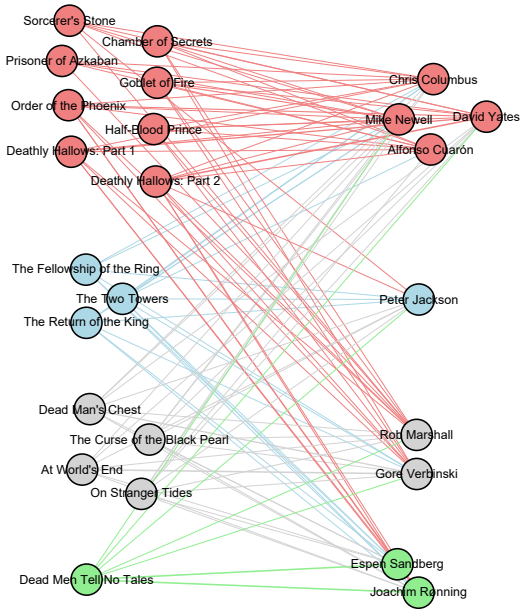
Meta-cesta MAG Táto meta-cesta predstavuje váženú maticu R^A_{MG}, ktorá je reprezentovaná bipartitným grafom. Táto sieť hovorí ako často herci jednotlivých filmov účinkujú v žánroch, do ktorých sa filmy radia. Váhy hrán nezobrazujeme kvôli prehľadnosti. Ako bolo uvedené vyššie,

film Dary smrti II vytvoril vlastný zhluk. Preto sme spustili algoritmus Co-Clustering (4, 5, 6) nad týmto bipartitným grafom a vo výsledku môžme vidieť (Obr. 25), že spomínaný film ako jediný patrí do zhluku mysterióznych filmov (Obr. 23).



Obr. 25: Meta-cesta MAG, algoritmus SCC (4, 5, 6), Q = 0,038

Meta-cesta MAD Táto meta-cesta predstavuje váženú maticu R^A_{MD}, ktorá je reprezentovaná bipartitným grafom. Hovorí ako často herci z jednotlivých filmov spolupracovali s režisérmi, ktorí tieto filmy režírovali. Váhy hrán nezobrazujeme kvôli prehľadnosti. Algoritmus spektrálne spoluzhlukovanie (4, 5, 6) detegoval zhluky, ktoré možno vidieť obrázku 26.



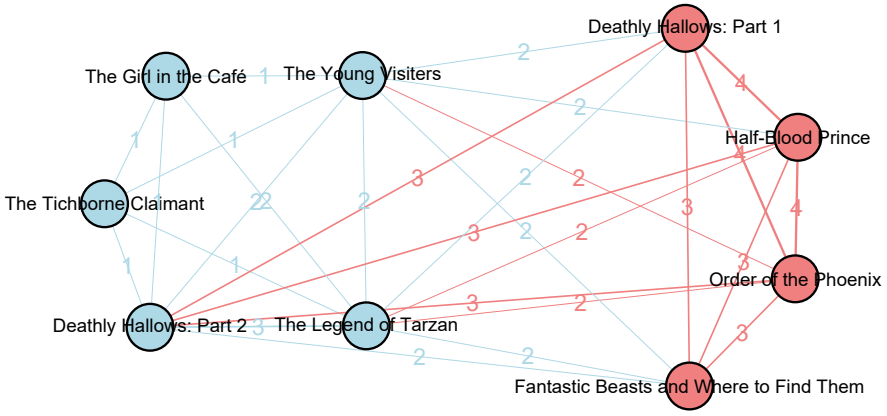
Obr. 26: Meta-cesta MAD, algoritmus SCC (4, 5, 6), Q = 0,535

6.3 Analýza bázovej sady filmov vybraných projekciou

Bázová sada obsahuje deväť filmov a bola vytvorená pomocou projekcie filmu Harry Potter a Fénixov rád a jeho režiséra. Konkrétne sa jedná o: Harry Potter a Fénixov rád, Harry Potter a Polovičný princ, Harry Potter a Dari smrti I, Harry Potter a Dari smrti II, Fantastické zvery a ich výskyt, Mladí návštevníci, Legenda o Tarzanovi, Dievča z kaviarne, Ja som Tichborne.

Projekcie na homogénne siete

Meta-cesta MGM Táto meta-cesta predstavuje váženú maticu R^G_{MM} filmov, ktoré patria do spoločných žánrov. Algoritmus Louvain v takejto sieti detegoval dva zhluky. Rovnakého výsledku dosiahlo aj spektrálne zhlukovanie s parametrami (2, 10, 2). Ostatné výsledky parametrov možno vidieť v tabuľke 6

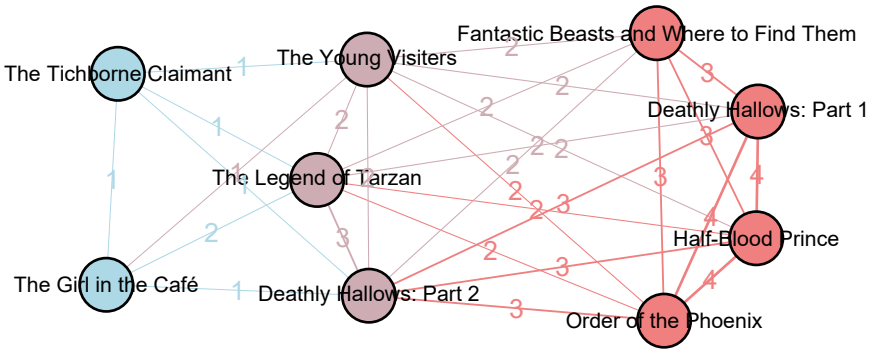


Obr. 27: Meta-cesta MGM, algoritmus Louvain, Q = 0,067

Tabuľka 6: Kvalita zhlukovania v MGM pre rôzne parametre SC, k-iterations = 100

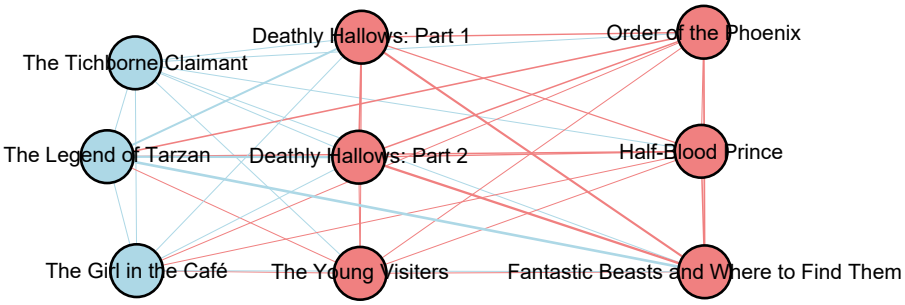
k-means / k-vectors	2	3	4	5	6	7	8	9
2	0,067	0,029	0,062	0,053	0,014	-0,046	-0,075	-0,035
3	0,010	0,028	0,031	-0,076	-0,021	0,012	-0,089	-0,100
4	-0,005	0,002	-0,025	-0,042	-0,037	-0,016	-0,072	-0,112
5	-0,005	-0,007	-0,030	-0,047	-0,033	-0,049	-0,079	-0,103

Algoritmus CPM s parametrom $k - cliques = 5$ rozdelil film na dva prekrývajúce sa zhluky cez filmy Harry Potter a Dary smrti II, Mladí návštevníci a Legenda o Tarzanovi. Fialová časť patrí k červenému zhluku ale aj k modrému.



Obr. 28: Meta-cesta MGM, algoritmus CPM (5), Q = 0,022

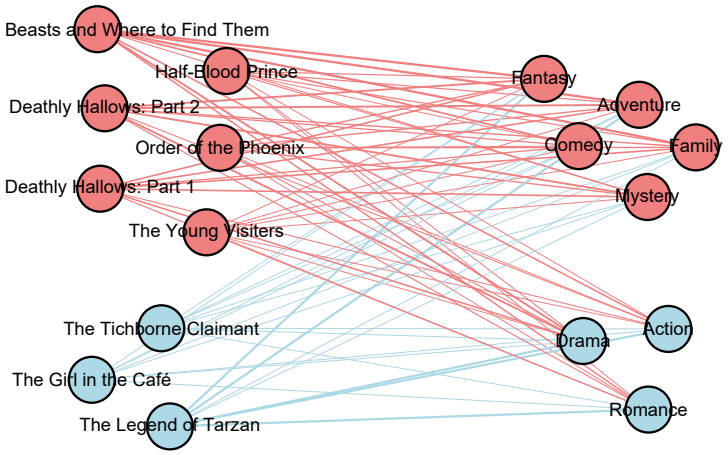
Meta-cesta MAGAM V tejto meta-ceste algoritmus Louvain detegoval iba jeden zhluk, avšak algoritmus SC (2, 5, 2) rozdelil filmy do dvoch zhlukov a výsledok môžeme porovnať s výsledkom zhlukovania nad meta-cestou MAG. Váhy hrán nie sú v obrázku zobrazené.



Obr. 29: Meta-cesta MAGAM, algoritmus SC (2, 5, 2), Q = -0,059

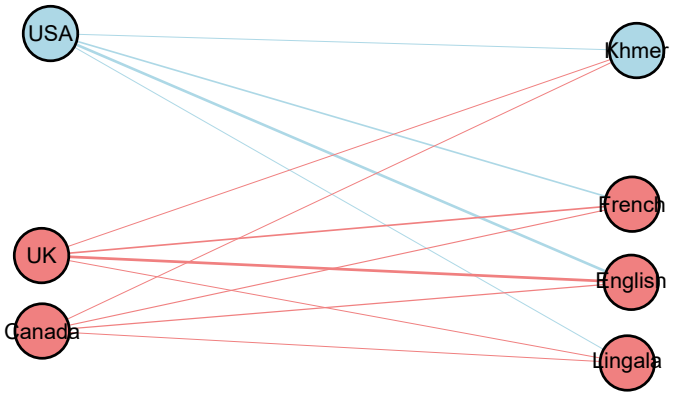
Heterogénne siete reprezentované bipartitným grafom

Meta-cesta MAG V tejto meta-ceste sme pomocou algoritmu SCC (2, 5, 4) určili do akých žánrov patria filmy v závislosti na hercoch. Pokiaľ by sme vzali iba reláciu R_{MG}, tak sieť popisujúca túto heterogénnu sieť je úplný bipartitný graf.



Obr. 30: Meta-cesta MAG, algoritmus SCC (2, 5, 4), Q = 0,054

Meta-cesta CAL Táto meta-cesta predstavuje váženú maticu R^A_{CL}, ktorá reprezentuje bipartitný graf. Tento graf má dva typy vrcholov: krajiny, ktoré spolu produkovali filmy z bázovej sady a jazyky, v ktorých bol film vydaný.



Obr. 31: Meta-cesta CAL, algoritmus SCC (2, 5, 6), Q = -0,030

7 Záver

Táto práca bola zameraná na problematiku heterogénnych sietí. Boli uvedené možnosti ich reprezentácie, príklady datasetov a analytických metód, ktoré sa v tejto problematike vyskytujú. Bola vybraná a spracovaná filmová databáza IMDb, ktorá predstavuje doménu analýzy. Boli predstavené meta-cesty, ich typy, sémantické významy a možnosti prevodov týchto meta-ciest na vážené matice susednosti, ktoré ďalej reprezentujú siete.

Ako hlavný analytický nástroj bola vybraná zhluková analýza. Boli zvolené algoritmy, ktoré zhlukujú na základe rôznych princípov a detegujú prekrývajúce sa ale aj neprekrývajúce sa zhluky. Medzi algoritmy, ktoré detegujú neprekrývajúce zhluky patria algoritmy: Louvain, spektrálne zhlukovanie, spektrálne spoluzhlukovanie. Pri spektrálnych algoritmoch je kritická voľba parametrov. K voľbe parametru $k - means$ je pri väčších dátach možné odhadnúť počet zhlukov pomocou metódy *eigengap* [35]. Keďže dáta nad ktorými sme prevádzali experimenty boli malé, tento prístup sa nejavil príliš vhodne. K vyhodnocovaniu kvality zhlukovania bola použitá modularita. Algoritmom, ktorý deteguje prekrývajúce sa zhluky je K-Clique Percolation, kvalita zhlukovania tohto algoritmu je taktiež vyhodnocovaná pomocou modularity.

Následne vznikla aplikácia, ktorá umožňuje používateľovi interaktívne pracovať s databázou a filtrovať dáta tak, aby pracoval s menším počtom dát, ktoré sú však vždy relevantné v zmysle analýzy. Okrem výberu dát umožňuje nad týmito dátami vytvárať rôzne meta-cesty s rôznou sémantikou, ktorú si používateľ sám volí (napr. pomocou meta-ciest AMA a $AMDMA$). Meta-cesty následne prevádzať na vážene matice a reprezentovať ich buď ako homogénne siete, alebo heterogénne siete, reprezentované bipartitnými grafmi s dvoma typmi vrcholov. Aplikácia umožňuje nad týmito sieťami spustiť zhlukovacie algoritmy a exportovať ich výsledky do formátu `GEXF`, ktorý je možné načítať v programe Gephi a vizualizovať tak výsledky zhlukovania.

Na záver boli predstavené experimenty, ktoré demonštrovali výsledky analýzy sietí z IMDb. Bázové sady k týmto experimentom sme vyberali tak, aby bolo jednoduché overiť zmysluplnosť výsledkov. Algoritmy boli testované s rôznymi variáciami vstupných parametrov a ich výsledky sme interpretovali na základe znalostí o bázovej sade. Aj keď algoritmus Louvain maximalizuje modularitu, tak spektrálne zhlukovanie dopadlo kvalitatívne rovnako. Ponúka sa možnosť pokračovať v tejto téme a rozšíriť počet metód k určovaniu kvality zhlukovania. Čo nebolo primárnym cieľom tejto diplomovej práce. Je však možné, že pri vyhodnocovaní kvality zhlukovania inou metódou by sa niektoré algoritmy javili vhodnejšie. Všetky experimenty sú exportované a súčasťou prílohy.

Citácie

[1] Jiawei Han. "Mining heterogeneous information networks by exploring the power of links". In: *International Conference on Discovery Science*. Springer. 2009, s. 13–30.

[2] Chuan Shi et al. "A survey of heterogeneous information network analysis". In: *IEEE Transactions on Knowledge and Data Engineering* 29.1 (2017), s. 17–37.

[3] Yizhou Sun a Jiawei Han. "Mining heterogeneous information networks: principles and methodologies". In: *Synthesis Lectures on Data Mining and Knowledge Discovery* 3.2 (2012), s. 1–159.

[4] Changping Meng et al. "Discovering meta-paths in large heterogeneous information networks". In: *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee. 2015, s. 754–764.

[5] Yizhou Sun et al. "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks". In: *Proceedings of the VLDB Endowment* 4.11 (2011), s. 992–1003.

[6] Erheng Zhong et al. "Modeling the dynamics of composite social networks". In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2013, s. 937–945.

[7] Bo Long, Zhongfei Mark Zhang a Philip S Yu. "Co-clustering by block value decomposition". In: *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM. 2005, s. 635–640.

[8] Inderjit S Dhillon. "Co-clustering documents and words using bipartite spectral graph partitioning". In: *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2001, s. 269–274.

[9] Mohsen Jamali a Laks Lakshmanan. "HeteroMF: recommendation in heterogeneous information networks using context dependent factor models". In: *Proceedings of the 22nd international conference on World Wide Web*. ACM. 2013, s. 643–654.

[10] Chuan Shi et al. "HeteRecom: a semantic-based recommendation system in heterogeneous networks". In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2012, s. 1552–1555.

[11] Honglei Zhuang et al. "Mining query-based subnetwork outliers in heterogeneous information networks". In: *Data Mining (ICDM), 2014 IEEE International Conference on*. IEEE. 2014, s. 1127–1132.

[12] Chuan Shi et al. "Ranking-based Clustering on General Heterogeneous Information Networks by Network Projection". In: *CIKM*. 2014.

[13] Chuan Shi a S Yu Philip. *Heterogeneous information network analysis and applications*. Springer, 2017.

[14] Yang Yang et al. “Predicting links in multi-relational and heterogeneous networks”. In: *Data Mining (ICDM), 2012 IEEE 12th International Conference on*. IEEE. 2012, s. 755–764.

[15] Deng Cai et al. “Community mining from multi-relational networks”. In: *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer. 2005, s. 445–452.

[16] Mark EJ Newman. “The structure and function of complex networks”. In: *SIAM review* 45.2 (2003), s. 167–256.

[17] Pavla Dráždilová, Jan Konečný a Miloš Kudělka. “Chain of Influencers: Multipartite Intra-community Ranking”. In: *International Computing and Combinatorics Conference*. Springer. 2017, s. 603–614.

[18] Yizhou Sun et al. “Rankclus: integrating clustering with ranking for heterogeneous information network analysis”. In: *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*. ACM. 2009, s. 565–576.

[19] Yizhou Sun, Yintao Yu a Jiawei Han. “Ranking-based clustering of heterogeneous information networks with star network schema”. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2009, s. 797–806.

[20] Ming Ji et al. “Graph regularized transductive classification on heterogeneous information networks”. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2010, s. 570–586.

[21] Ming Ji, Jiawei Han a Marina Danilevsky. “Ranking-based classification of heterogeneous information networks”. In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2011, s. 1298–1306.

[22] Bokai Cao, Xiangnan Kong a S Yu Philip. “Collective prediction of multiple types of links in heterogeneous information networks”. In: *Data Mining (ICDM), 2014 IEEE International Conference on*. IEEE. 2014, s. 50–59.

[23] Lawrence Page et al. *The PageRank citation ranking: Bringing order to the web*. Tech. spr. Stanford InfoLab, 1999.

[24] Jon M Kleinberg. “Authoritative sources in a hyperlinked environment”. In: *Journal of the ACM (JACM)* 46.5 (1999), s. 604–632.

[25] Ding Zhou et al. “Co-ranking authors and documents in a heterogeneous network”. In: *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE. 2007, s. 739–744.

[26] Michaek Kwok-Po Ng, Xutao Li a Yunming Ye. “Multirank: co-ranking for objects and relations in multi-relational data”. In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2011, s. 1217–1225.

[27] Chuan Shi et al. “Semantic path based personalized recommendation on weighted heterogeneous information networks”. In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM. 2015, s. 453–462.

[28] Yehonatan Cohen, Danny Hendler a Amir Rubin. “Node-centric detection of overlapping communities in social networks”. In: *International Conference and School on Network Science*. Springer. 2017, s. 1–10.

[29] Vincent D Blondel et al. “Fast unfolding of communities in large networks”. In: *Journal of statistical mechanics: theory and experiment* 2008.10 (2008), P10008.

[30] Gergely Palla et al. “Uncovering the overlapping community structure of complex networks in nature and society”. In: *nature* 435.7043 (2005), s. 814.

[31] Doc RNDr Petr Hliněný. “Základy Teorie Grafu”. In: *Brno: Masarykova Univerzita* (2010).

[32] Coen Bron a Joep Kerbosch. “Algorithm 457: finding all cliques of an undirected graph”. In: *Communications of the ACM* 16.9 (1973), s. 575–577.

[33] Etsuji Tomita, Akira Tanaka a Haruhisa Takahashi. “The worst-case time complexity for generating all maximal cliques and computational experiments”. In: *Theoretical Computer Science* 363.1 (2006), s. 28–42.

[34] Andrew Y Ng, Michael I Jordan a Yair Weiss. “On spectral clustering: Analysis and an algorithm”. In: *Advances in neural information processing systems*. 2002, s. 849–856.

[35] Ulrike Von Luxburg. “A tutorial on spectral clustering”. In: *Statistics and computing* 17.4 (2007), s. 395–416.

8 Príloha na CD/DVD

Teoretická časť - /doc:

- Práca.pdf

Praktická časť - /src:

- /experiments - Gephi súbory z experimentov
- /mysql - SQL skript s databázou IMDb
- /solution - zdrojový kód